Toward Human-Inspired AI: Identifying Data, Building Structures, and Hypothesis-Driven Learning

by

Stéphane Aroca-Ouellette

B.A.Sc., University of Toronto, 2018

M.S., University of Toronto, 2020

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2025

Committee Members:

Prof. Alessandro Roncone, Chair

Prof. Katharina von der Wense, Chair

Prof. Bradley Hayes

Prof. Maria Pacheco

Dr. Rin Susa Metcalf

Aroca-Ouellette, Stéphane (Ph.D., Computer Science)

Toward Human-Inspired AI:
Identifying Data, Building Structures, and Hypothesis-Driven Learning

Thesis directed by Prof. Alessandro Roncone and Katharina von der Wense

Artificial intelligence (AI) systems frequently fail to generalize effectively to novel and out-of-distribution scenarios, fundamentally due to their sole reliance on extensive, data-driven learning. Such methods are known to exploit surface-level correlations rather than learn deeper causal structures, leading to significant issues in scenarios marked by data scarcity or novelty.

In contrast, humans excel in generalization through causal reasoning, efficiently adapting existing knowledge and continuously refining it through hypothesis-driven learning. This thesis investigates how core human cognitive mechanisms—specifically data selection, structural abstraction, and hypothesis-driven learning—can inspire algorithmic advancements to address AI's generalization limitations.

First, we demonstrate the importance of comprehensive multi-modal data streams, showing that richer, contextually grounded data enhances generalization in natural language understanding and human-agent collaboration. Next, we explore structured representations by proposing a hierarchical reinforcement learning framework mirroring human cognitive structures, significantly improving agent adaptability in human-agent collaboration. Finally, we introduce PROSE, a hypothesis-driven learning method enabling AI models to rapidly infer and iteratively refine latent user preferences from limited data.

Collectively, this thesis underscores the potential of human-inspired methodologies to create AI systems that not only generalize more robustly but are inherently better aligned with human norms and expectations, paving the way toward truly adaptive, human-centered artificial intelligence.

Acknowledgements

I would like to begin by expressing my deepest gratitude to my committee members—Alessandro Roncone, Katharina von der Wense, Bradley Hayes, Maria Pacheco, and Rin Susa Metcalf—for their guidance, feedback, and support throughout this journey. Their insights and encouragement have been invaluable in shaping both this dissertation and my development as a researcher.

I am equally grateful to the members of the HIRO lab and the NALA lab, whose ideas, collaborations, and camaraderie made the research process both stimulating and rewarding.

To my family, thank you for your constant love and support. In particular, I wish to thank my parents, Patricia Aroca-Ouellette and Patric Ouellette, for instilling in me perseverance, curiosity, and a commitment to learning.

Finally, to my wife, Justine Hamilton-Arvisais. Words cannot fully capture the depth of my gratitude. You kept me grounded through every stage of this work—feeding me when I forgot to eat, holding our life together when my focus was elsewhere, and providing endless encouragement when the challenges felt overwhelming. This dissertation would not exist without your patience, love, and unwavering belief in me. I am profoundly lucky to share my life with you.

Contents

Chapter

1 Introduction **1**

2 Background **6**

2.1 The Role of Data in Generalization 6

2.2 The Role of Structure in Generalization 7

2.3 The Role of Learning Mechanisms in Generalization 9

3 Generalization through Data **11**

3.1 PROST: Physical Reasoning about Objects through Space and Time 12

3.1.1 Introduction 12

3.1.2 Related Work 14

3.1.3 PROST 17

3.1.4 Models 19

3.1.5 Results 23

3.1.6 Analysis 24

3.1.7 Discussion 27

3.1.8 Conclusion 29

3.2 The World of an Octopus: How Reporting Bias Influences a Language Model's Perception of Color 29

3.2.1 Introduction 29

3.2.2 CoDa . . . 31
3.2.3 Reporting Bias . . . 36
3.2.4 Experimental Setup . . . 37
3.2.5 Results . . . 41
3.2.6 Limitations . . . 44
3.2.7 Related Work . . . 45
3.2.8 Conclusion . . . 46
3.3 ReSeeding Latent States for Sequential Language Understanding . . . 47
3.3.1 Introduction . . . 47
3.3.2 Related Work . . . 49
3.3.3 Method . . . 51
3.3.4 Research Questions . . . 57
3.3.5 Results & Discussion . . . 58
3.3.6 Conclusion . . . 61
3.3.7 Limitations . . . 62
3.4 Eyes on the Game: Deciphering Implicit Human Signals to Infer Human Proficiency, Trust, and Intent . . . 63
3.4.1 Introduction . . . 63
3.4.2 Related Work . . . 66
3.4.3 Method . . . 67
3.4.4 Models . . . 73
3.4.5 Experimental Design . . . 74
3.4.6 Results . . . 75
3.4.7 Discussion and Conclusion . . . 78
4 Generalization through Structures **80**
4.1 Implicitly Aligning Humans and Autonomous Agents through Shared Task Abstractions . . 81

4.1.1 Introduction 81
4.1.2 Related Works 83
4.1.3 Method 85
4.1.4 Experimental Design 87
4.1.5 Results 92
4.1.6 Discussion 96

5 Generalization by Building Theories **98**

5.1 Aligning LLMs by Predicting Preferences from User Writing Samples 98
5.1.1 Introduction 99
5.1.2 Related Work 100
5.1.3 PROSE 101
5.1.4 Assistive Writing Benchmark 103
5.1.5 Experimental Set Up 108
5.1.6 Results and Discussion 110
5.1.7 Conclusion 113

6 Discussion and Conclusion **116**

6.1 Discussion 116
6.2 Future Work 117
6.3 Conclusion 118

Bibliography **120**

Appendix

A Author Contributions for Presented Papers **161**

A.1 PROST 161

A.2 The World of an Octopus 161

A.3 RESEED 162

A.4 Eyes on the Game 162

A.5 HA2 163

A.6 PROSE 163

B Appendix for The World of an Octopus **165**

B.1 Dataset Construction 165

B.2 Analysis of Annotator Biases 165

B.3 Experimental Details 165

  B.3.1 Representation Probing 166

B.4 Zero Shot Results 166

C Appendix for RESEED **168**

C.1 HOUSE details 168

C.2 ICL prompt 170

C.3 Two-Step Training 172

C.4 Additional Implementation Details 172

  C.4.1 Hyper parameters 172

  C.4.2 Computational Budget 173

C.5 AI assistant use 173

C.6 Additional Attributions and Attribution Info 173

D Appendix for PROSE **174**

D.1 Algorithm 174

D.2 Metric Definitions 176

  D.2.1 Preference Inference Quality 176

  D.2.2 Generation Quality 176

D.3 Extended Results 179

D.3.1 Metric Correlation 179

D.3.2 Baselines and Ablations per LLM 180

D.3.3 PRELUDE Results 183

D.3.4 Preference Inference and Conditioning Performance by Number of User Samples . . 185

D.4 PRELUDE vs. PLUME Preference Sets 186

D.5 Illustrative Examples of Issues with PRELUDE 188

D.5.1 Levenshtein Distance and Multiple Generations 188

D.5.2 Editing Influences the User 190

D.6 Prompts 192

D.6.1 Preference Inference and Preference-Conditioned Agent Prompts 192

D.6.2 Synthetic Human Prompts 197

D.6.3 Preference-Conditioned Agent Baseline Prompts 199

D.6.4 Qualitative Verification Consistency Examples 201

D.6.5 Qualitative Iterative Refinement Examples 203

Chapter 1

Introduction

For much of its history, artificial intelligence (AI) research has focused on narrow, well-defined tasks—such as classifying images or mastering games. As AI systems have grown increasingly powerful, there is mounting interest in developing more general and adaptable agents. However, initial progress in this direction has underscored the profound complexity of this challenge. In this thesis, I argue that overcoming this challenge requires turning to ourselves: **humans remain the gold standard for generalizable intelligence, providing invaluable insights for addressing the central challenge of generalization.**

The development of artificial intelligence has long been driven by the pursuit of generalization. Early systems relied on manually defined rules to solve highly constrained tasks [92, 99], but these brittle rule-based approaches quickly failed when faced with novel or unexpected inputs. The advent of machine learning marked a transformative shift; models could now be trained from data rather than hand-coded logic [197, 189]. This paradigm greatly expanded the scope of solvable problems and led to models that perform well across diverse inputs, as long as they remain within the distribution of the training data. Fueled by this success, modern AI has entered an era of rapid scaling, exemplified by the proliferation of massive models and datasets and designed to capture increasingly large distributions [294, 295, 73].

Yet, this data-driven approach is reaching a series of fundamental limitations. Perhaps most evidently, there are situations where it is impractical or infeasible to collect data. Inherently, novel situations—such as scientific discoveries, new political events, or interacting with someone you have never met before—do not come with datasets to train on. Even in domains where data collection is possible, the long-tail nature of real-world distributions poses significant challenges [147]. Autonomous driving is a striking example.

Although high-quality driving data has been collected since at least 2012 [113], data capturing rare but critical events—such as high-speed crashes or near misses—is difficult, costly, and potentially dangerous to collect. These rare cases, however, are precisely the ones that autonomous systems must master to ensure safety and reliability. The scarcity of such data is a key reason the deployment of autonomous vehicles has remained "two years away" for over a decade [9].

This limitation stems directly from the model's training. Data-driven machine learning systems learn entirely through small stochastic optimization steps, meaning that meaningful and robust learning will require a large number of samples that scales with the size, and in turn performance, of the model [145]. Further, they learn to rely on statistical correlations rather than true causal understanding, leading to what has been termed "shortcut learning" [114]. When these correlations are spurious, models frequently produce seemingly inexplicable and absurd errors when deployed. For example, a classifier trained to distinguish huskies from wolves was trained using pictures in which all the huskies were in snow, whereas the wolves were not. This led the model to predict the class based solely on the presence of snow rather than the visual characteristics of the animal [309]. In another case, a model trained to predict pneumonia outcomes mistakenly inferred that asthma patients were at lower risk than those without [59]. Of course, asthmatic patients are actually at a higher risk when infected with pneumonia. This leads to them being treated more aggressively by doctors, creating the correlation in the data that was picked up by the model. In both examples, the models failed to uncover the underlying causal structure and instead rely on surface-level correlations, leading models to produce high-probability outputs according to the training distribution, rather than factually or causally correct ones [335, 403]. The result is significant degradation in performance under distribution shift.

These failures are particularly acute in language models. Despite their high level of fluency and seemingly coherent outputs, language models often generate factually incorrect statements [27, 239]. These 'hallucinations' compound and are exacerbated in tasks requiring multi-step reasoning, leading to shockingly poor performance in tasks such as compositional logic [91], planning [379], or length generalization [407]. This pattern reflects a deeper issue: the lack of structured representations founded in causal understanding prevents the effective adaptability needed to reason in out-of-distribution situations.

Humans, in contrast, excel at generalization. We can adapt known causal priors to succeed in entirely new environments, quickly and continuously adapt from a handful of examples or demonstrations to improve in these environments, and build complex knowledge structures to further generalize and optimize our actions. A core enabler of this generalization is our causal understanding of the world. Rather than passively absorbing patterns, from a young age humans utilize their prior knowledge of related concepts to build causal theories of the environment [134, 126], and then interact with the environment to test and refine these causal models [329, 127]. This causal understanding underpins not only our ability to reason and act in the world [346], but is also critical to our capacity to identify salient data for the task at hand [46]; build modular, transferable abstractions [224]; and to create new theories that can be tested to further build our causal understanding[125].

Indeed, we argue that this causal understanding is what enables humans to generalize where modern AI systems falter. When faced with novelty, long-tailed situations, or data scarcity, rather than relying on surface correlations, humans draw on past experience to identify the relevant variables, build robust and generalizable structures built and extensively tested through lived experience and communication with others. In these unfamiliar settings, humans continue to formulate and evaluate new causal theories to expand our understanding and adapt to the domain at hand. Inherently, language plays a critical and symbiotic role in this endeavor: it provides a flexible framework to structure and refine causal theories [346], while meaningful language understanding depends on the causal knowledge built through environmental interaction [37, 312].

Given the clear strengths of human cognition in generalization, **this thesis explores how we can design more generalizable AI systems by drawing inspiration from the way humans learn, reason, and adapt**. I focus on three core mechanisms underlying human generalization—data selection, structural abstraction, and hypothesis-driven learning—and investigate how these can be translated into algorithmic frameworks.

First, in Chapter 3, we begin by demonstrating the importance of providing AI systems with data that offers a comprehensive and cohesive representation of the task environment. We specifically investigate data streams that humans rely on to capture the salient signals for effective generalization. Because data provides the foundational signals from which systems learn, its nature and scope fundamentally shape what is learned. Sparse coverage, class imbalances, or the absence of critical edge cases can hinder a system's ability to distinguish between spurious correlations and meaningful, generalizable patterns. We identify two domains, natural language understanding and zero-shot coordination, where humans naturally draw on rich, multi-modal data streams that are often absent from deep learning pipelines. In human language learning, interactive experiences with the physical world are essential to ground concepts [37]. In contrast, large language models (LLMs) are typically trained on isolated text, lacking the environmental context required to form generalizable representations [34]. Similarly, in collaborative tasks, humans use non-verbal cues such as gaze and gesture [2]to infer the intentions of others. These are signals that are rarely exploited in current collaborative AI systems. In both cases, we show that incorporating analogous data modalities into AI training improves generalization, suggesting that the data sources humans leverage offer valuable guidance for building more adaptive models.

Second, in Chapter 4, we investigate the role of structure in supporting generalization. We define structure broadly as the organization of how information flows through a system. Due to their scalability and flexibility, modern deep learning systems are typically built around the architecture of neural networks. However, alternative learning paradigms, such as Bayesian belief networks [275] and random forests [48], have been proposed to explicitly embed structural constraints in order to guide inference. Most high-performing systems today also incorporate higher-level structures, such as modular components or temporal abstractions. Despite this progress, current AI systems still fall short of the flexibility and generality with which humans construct and apply structure. Humans fluidly organize knowledge using conceptual hierarchies, cognitive frameworks [194], and taxonomies [256], enabling efficient reuse, reduced search, and rapid adaptation. This gap is particularly pronounced in collaboration: while humans rely on shared abstractions to align goals and coordinate behavior, most AI systems lack the means to represent or reason over such shared structure. Motivated by this, we propose a hierarchical reinforcement learning framework for human-agent collaboration that incorporates shared task abstractions. By enabling agents to operate across multiple levels of abstraction, we show that this approach supports more natural interaction with human partners and improves coordination in collaborative settings.

Finally, in Chapter 5, we investigate learning mechanisms, specifically how systems update their internal representations in response to data. Most modern machine learning models rely on gradient-based optimization over large datasets, gradually adjusting parameters through many small updates. In contrast, humans can rapidly adapt from limited experience by forming and testing high-level hypotheses about the underlying causes of observed behavior [126]. Inspired by this, we develop PROSE, a method that infers latent user preferences about writing style from writing samples, then iteratively refines and verifies these inferred preferences across previously observed examples. This hypothesis-driven learning process improves the precision of inferred preferences and enables more effective alignment between model behavior and the user it aims to imitate.

Collectively, this thesis motivates and demonstrates how insights from human generalization can be leveraged to develop AI agents that can adapt and transfer what they learn in more robust meaningful ways. By drawing directly from how humans learn, reason, and interact, these systems become not only more capable but also more naturally aligned with human norms, expectations, and goals. In this way, human-inspired design provides a direct path toward more human-centered AI: agents that operate in ways that are interpretable, appropriate, and effective when interacting with humans.

Chapter 2

Background

Causal understanding has long been a central topic in both human cognition [346, 286] and machine intelligence [141]. Despite progress in both domains, how to replicate the breadth and flexibility of human causal reasoning, and more broadly, the human capacity for knowledge and skill generalization, remain open questions. It is increasingly clear, however, that any such progress will depend critically on three components: (1) the data available to the system, (2) the structure of the system, and (3) the learning mechanisms it employs. In this chapter, we review related work in each of these areas.

2.1 The Role of Data in Generalization

In the introduction, we highlighted how spurious correlations in data—such as snow in husky vs. wolf images [309] or asthma in pneumonia patient records [59]—can cause brittle models that generalize poorly. In both cases, the models latched onto statistically predictive but semantically irrelevant features. Notably, adding counterexamples that broke these correlations improved generalization, illustrating how the nature of the training data fundamentally shapes model behavior.

This observation extends across domains and modalities. In 3D object recognition, PointNet [291] shows that using point clouds leads to better generalization than voxels or 2D projections, likely because point clouds preserve spatial structure more faithfully. In robotic control, Play-LMP [233] demonstrates that "play data" which is collected from natural, undirected interactions covers a broader action distribution than expert demonstrations or random actions, resulting in policies that are more robust to perturbations. In emotion recognition, PDCNN [369] finds that models trained on combined facial, textual, and vocal inputs outperform unimodal baselines, highlighting the importance of integrating complementary cues. Similarly, Flex-Judge [183] shows that language models trained on diverse human reasoning patterns generalize better across modalities and tasks, even with fewer examples.

Together, these studies suggest that effective generalization depends not just on the quantity or diversity of data, but on its alignment with the informational structures humans rely on: richly descriptive, task-relevant, and reflective of the causal or compositional signals that underlie good decision-making. When training data mirrors the multimodal, high-coverage, and semantically meaningful experiences that humans learn from, models can achieve stronger and more transferable generalization.

2.2 The Role of Structure in Generalization

Whereas data defines the inputs and outputs of a system, structure defines how information flows through it. Among structural approaches, hierarchies—structures that organize components across multiple levels of abstraction—are particularly prominent in both human cognition [10] and artificial agents [382], and have long been recognized for their ability to support generalization and efficiency [30]. These benefits are often attributed to two key properties: *feature reuse* and *sparsity*. *Feature reuse* enables components learned in one context to be applied in others, while *sparsity* constrains the search space and can help prevent the model from relying on irrelevant correlations. For example, decomposing images into background and foreground could prevent a model from learning to associate "snow" with "husky." Together, these properties provide a powerful foundation for efficient and robust adaptation.

In modern deep learning, the dominant structural foundation is the architecture of neural networks. While these systems have the flexibility to learn directly from data, they are typically less structured than human-defined systems. Nevertheless, many high-performing neural architectures exploit **architectural structure** to achieve generalization through *feature reuse* and *sparsity*. For example, convolutional neural networks (CNNs) [199] apply the same filters across spatial locations, enabling the reuse of local features like edges and textures throughout an image. This weight sharing reduces the total number of parameters and encourages the model to learn compact, transferable representations. Similarly, transformers [381] apply a shared attention mechanism across all input tokens, allowing the model to detect recurring patterns—such as syntactic or semantic dependencies—regardless of their position. In both architectures, the repeated application of a fixed set of operations introduces sparsity and facilitates generalization by focusing computation on patterns that appear across contexts. Other works push sparsity more explicitly: DreamerV2 [136] uses discrete latent representations to constrain the representational space, while LISA [111] improves generalization by introducing a hierarchical bottleneck enforced through a codebook, which is shown to be particularly effective in low-resource settings.

Beyond architectural structure, many systems now incorporate explicit *task*, *behavioral*, or *spatial hierarchies*, often defined by humans to scaffold learning and decision-making. Hierarchical reinforcement learning (HRL) [384], for example, uses human-specified abstractions to organize and sequence lower-level policies, enabling agents to operate across multiple temporal and conceptual levels. SayCan [161] builds on this idea by combining a library of independently learned robotic skills with a high-level planner that selects actions using a pre-trained language model to interpret intent and an RL-trained value function to assess feasibility. Similarly, HULC++ [243] employs a spatial hierarchy: a model-based controller handles coarse, long-horizon planning, while a learning-based policy manages fine-grained motion near the goal. These systems demonstrate how injecting human-defined structure into deep learning models can improve both generalization and transparency. Yet even with these advances, neural networks remain heavily overparameterized. Research shows that only 10–20% of parameters are needed to maintain performance after training [105], However, identifying these components and their structure a priori remains a major challenge.

Humans, in contrast, can often use their broad knowledge and causal understanding to identify relevant concepts in order to build sparse and targeted *conceptual hierarchies* [372]. These structures evolve with experience and enable humans to generalize quickly, even in unfamiliar environments. Crucially, they span multiple levels of abstraction—from low-level sensorimotor patterns to high-level conceptual groupings—allowing humans to flexibly organize and apply knowledge across diverse tasks.

Interestingly, recent work has shown that very large pre-trained networks exhibit some degree of sparse concept encoding. For example, OpenAI was able to identify “multi-modal neurons” that would activate for the same concept, regardless if the concept was shown as an image or text [122]. Anthropic analyzed the internal “biology” of an LLM and were able to parse subsets of neurons, i.e., interpretable building blocks, that consistently fired for specific concepts. By tracing the interactions among these features, they observed patterns that resembled planning, multi-hop reasoning, and solving problems using multiple pathways [217]. While current work remains focused on identifying such structures, developing methods to systematically harness these pathways may be a promising direction in the future for more targeted updates.

2.3 The Role of Learning Mechanisms in Generalization

In this section, we investigate how a system updates its internal representations in response to data. Most modern deep learning systems rely on variants of stochastic gradient descent [43], which iteratively adjust model parameters by predicting outputs, computing the loss with respect to ground truth, and backpropagating gradients to minimize that loss. While this mechanism enables learning directly from raw data without manual supervision and exhibits predictable scaling laws [171], it often requires massive amounts of data and compute, and in turn money, to train performant systems. To improve efficiency, various strategies have been developed to reduce the data burden. These include data augmentation via jittering and transformation [189], the use of latent imagination to simulate trajectories and outcomes [136], and more recently, synthetic data generation using large models to bootstrap training pipelines [21]. However, these approaches primarily mitigate the inefficiencies of gradient-based learning rather than addressing their root causes.

In contrast, humans are able to adapt quickly from just a few examples, interactions, or instructions. A core hypothesis for this capability is the *theory theory* [125], which suggests that humans learn by forming high-level hypotheses about the world, testing them, and updating their beliefs based on the outcomes [126]. Notably, these updates occurs at a conceptual level, rather than at the level of individual neurons [372]. This results in larger, more targeted “updates”. Further, because the human directly acts on the world instead of passively observing it, it is easier to infer the causal effect of their action.

Meta-learning, or “learning to learn,” seeks to narrow this gap by enabling models to internalize more effective update strategies from prior tasks. Algorithms like MAML [100] and Reptile [261] train models to be highly adaptable with just a few gradient steps, essentially optimizing for rapid generalization. These approaches mirror aspects of human learning, where experience with prior tasks shapes the learner’s inductive biases, allowing new tasks to be learned more efficiently. However, most meta-learning methods remain fundamentally gradient-based, require inefficient inner-loop optimization, and struggle to scale beyond simplified tasks or curated benchmarks.

While still in its infancy, recent research has begun exploring models that close the gap to human by learning by endowing them with the ability to generate and test hypotheses [198]. Modern LLMs, equipped with broad prior knowledge from pre-training and the ability to reason in and through natural language, can form hypotheses, test them, and extract insights to guide future behavior. This paradigm has been applied to emulate human writers [109], align robot behavior with human preferences [277], and, most notably, to achieve a gold medal at the International Math Olympiad [157]. While promising, such methods remain constrained by the representational limits of language and the fixed nature of pretraining. Moreover, despite their scale, current models still fall short of human-level causal understanding and grounded reasoning.

Chapter 3

Generalization through Data

In this chapter, I present my work on leveraging insights from how humans select and utilize data to enhance generalization in AI systems. I explore this theme through two distinct domains: language understanding and human-agent interaction.

In the first part of this chapter, I examine several approaches to grounding language. Modern large language models (LLMs), such as GPT-4 [267], demonstrate that language possesses exceptional generalizability due to its flexibility in encoding shared human knowledge, supporting abstraction, and enabling unconstrained reasoning across varied contexts. However, human language acquisition research emphasizes that meaningful linguistic understanding is fundamentally rooted in embodied interactions with the physical environment [115, 37, 139, 120, 108], a critical component largely absent from current text-only LLM training. My initial probing (Section 3.1) highlights significant shortcomings of LLMs when faced with physical reasoning tasks. Follow-up work (Section 3.2) identifies reporting bias—the human tendency to omit information perceived as obvious—as a contributing factor to these limitations, and demonstrates that multi-modal approaches better mitigate this issue. To address this gap directly, I introduce RESEED (Section 3.3), a novel method integrating environmental data into LLM training, significantly improving performance on challenging out-of-domain sequential reasoning tasks.

In the second part of this chapter (Section 3.4), I explore methods for rapidly identifying and interpreting implicit data signals from human teammates. Effective human-agent interaction depends crucially on quickly understanding key attributes of teammates, including their proficiency, trustworthiness, and intent. Humans excel at extracting such information by implicitly attending to behavioral cues such as actions, body language, and eye gaze when explicit communication is limited or impossible [106]. Here, I investigate how agents can similarly leverage behavioral patterns and eye gaze data to predict these teammate characteristics. My findings reveal that both behavior and gaze independently provide valuable predictive signals, but importantly, their integration—a natural cognitive process for humans—achieves significantly stronger and more reliable inference. These continuous implicit signals thus represent an essential data stream for agents aiming to rapidly adapt and perform effectively alongside new teammates.

3.1 PROST: Physical Reasoning about Objects through Space and Time

The work described in this section has been published in ACL-Findings 2021 [15].

3.1.1 Introduction

In the context of natural language processing (NLP), [28] provides a working definition of "understanding" as the ability to recover the communicative intent from an utterance. To achieve this, one must be able to query a set of concepts that is aligned with the speaker's own understanding. An example of such alignment is our interaction with the physical world. This experience, shared by all humans, provides a common set of concepts to rely on in communication. For example, the reader can map the phrase **I dropped my pint glass** to a set of relevant experiences and generate a mental depiction of the scene. Further, the reader can also use their knowledge of gravity and the properties of a pint glass to reason about potential outcomes: the pint glass will fall toward the ground and will likely break on impact.

A person drops a *glass*, a *pillow*, a *coin*, and a *pen* from a balcony.
The [MASK] is most likely to break.
A) glass **B)** pillow **C)** coin **D)** pen

Figure 3.1: An example question from PROST.

Children grab, push, and play with the objects around them to form concepts about the world they live in even before learning to talk [139]. These concepts are then linked with words to enable communication, eventually providing the necessary grounds for concepts and language to co-develop [37, 115]. In contrast, current language models (LMs) are not exposed to real-world experiences, making them incapable of grounding language [33]. We hypothesize that this lack of experience impedes their ability to both understand an utterance relating to the physical world and their ability to reason about its implications.

In order to investigate our hypothesis, we create **PROST: Physical Reasoning of Objects Through Space and Time**, a probing dataset to evaluate the ability of pretrained LMs to understand and reason about the physical world. PROST consists of multiple-choice cloze-style questions covering 10 basic concepts: direction, mass, height, circumference, stackable, rollable, graspable, breakable, slideable, and bounceable. Importantly, PROST is designed to avoid models succeeding in unintended ways. First, PROST provides no training data, so as to probe models in a zero-shot fashion. This prevents models from succeeding through spurious correlations between training and test data and encourages success through a true understanding of and reasoning about the concepts at hand. Second, we manually write templates for all questions in an effort to prevent models from having seen the exact same sentences in their training data. Finally, it focuses on a small set of well defined, objective concepts that only require a small vocabulary. This allows researchers to focus more on the quality of training data rather than the size of it.

Contributions We make two contributions: 1) We introduce PROST, a dataset with $18,736$ cloze-style questions created from 14 manually written templates, covering 10 physical reasoning tasks. 2) We conduct an extensive analysis which demonstrates that state-of-the-art pretrained models are inadequate at physical reasoning. More specifically, they are influenced by the order in which answer options are presented to them, they struggle when the superlative in a question is inverted (e.g., *most* ↔ *least*), and increasing the amount of pretraining data and parameters only yields minimal improvements. The dataset and code is available at github.com/nala-cub/prost.

3.1.2 Related Work

Evaluation of Reasoning Abilities As pretrained models are excelling on many NLP tasks, more work is being done on understanding their abilities. A subset of this work focuses on physical reasoning. PIQA [35] tests physical commonsense, with concepts ranging from hard shell tacos to separating egg yolks. In order to succeed on PIQA through reasoning, a model would need to be able to understand thousands of human experiences. In contrast, PROST provides a first step towards grounded understanding and reasoning by focusing on a few simple concepts. [22] provides a set of 2D puzzles that involve placing a new object in a scene to accomplish a goal. This research also focuses on simple physics, however there is no language component. [71] and [174] both provide a large set of grade school multiple-choice questions, including some that could be solved with reasoning. However both provide corresponding material where the solution can be found, relying more on information retrieval than a general understanding and reasoning about the world.

Another set of reasoning-based benchmarks focuses on common sense reasoning. SWAG and its extension hellaSWAG evaluate commonsense natural language inference [414, 415]. [325] tests commonsense reasoning about social situations. However, commonsense reasoning is often subjective and requires understanding of complex human–human interactions involving social and societal norms. In contrast, physical reasoning is based on objective and well defined constructs.

Other datasets [101, 93, 121] focus on object–attribute comparison. However, they compare concepts at a word level rather than sentence level and use a large training set to create an engineered object–attribute comparison model. It is difficult to see how these models could generalize to other forms of reasoning.

Moreover, all the above datasets follow a pretraining-agnostic identically distributed (PAID) paradigm

[218], making them susceptible to models that can leverage unintended correlations between the training and test sets.

Zero-Shot LM Probes Similar to PROST, several recent benchmarks have circumvented the concern of identically distributed training and test sets by probing models in a zero-shot manner. [282] queries masked LMs (MLMs) for factual knowledge using templates in the format of **Dante was born in [MASK].** [363] use a similar format to probe six concepts ranging from age comparison to taxonomy conjunction. [97] uses this format to show that BERT robustly retrieves hypernyms, but fails to understand negation. [212] probe numerical commensense in both MLMs and traditional LMs. [397] measures traditional LMs' sense of grammatical acceptability by comparing sentence probabilities.

Grounded Language Environments PROST investigates if pretrained models show a lack of understanding of the physical world which could result from learning language without grounding. While not used for pretraining, a number of multi-modal environments have been developed to ground language. [338]'s ALFRED builds on other vision-and-language navigation environments [129, 308, 432, 8], and enables grounding of language instruction to actions, behaviours, and objects. BABYAI [66] and BABYAI++ [56] provide an environment to ground simple language in a gridworld. Additionally, other work has explored grounding language in simulations or the real world [140, 234]. While they provide important resources to ground language, little emphasis is placed on the language modules themselves. They are often trained tabulae rasae, learning language for a singular purpose and missing out on the syntax and coverage learnt during pretraining;[1] language is only ever an input, and no analysis has been done on how language understanding evolves as the agent learns to succeed on different tasks.

---

[1] An exception is [234], which incorporates modern LMs and provides impressive generalizability. However, they too only use language as an input and do not analyze how language understanding evolves.

Category	Qs.	Template
Directs. 1	12	**C:** A person is walking {*north/east/south/west*}. They turn {*left/right/around*}.
		Q: They are now walking [MASK].
		O: **A)** north **B)** east **C)** south **D)** west
Directs. 2a	1	**C:** A person drops a ball.
		Q: Immediately after leaving the person's hand, the ball is moving toward the [MASK].
Directs. 2b	1	**C:** A person throws a ball straight into the air.
		Q: Immediately after leaving the person's hand, the ball is moving toward the [MASK].
Directs. 2c	1	**C:** A person throws a ball straight into the air.
		Q: Immediately after reaching the highest point in it's trajectory, the ball is moving toward the [MASK].
Directs. 2d	1	**C:** A person drops a ball. The ball then bounces off the ground.
		Q: Immediately after bouncing off the ground, the ball is moving toward the [MASK].
		O: **A)** ground **B)** sky **C)** left **D)** right
Mass 1	720	**C:** A(n) {*mass_obj1*}, a(n) {*mass_obj2*}, a(n) {*mass_obj3*}, and a(n) {*mass_obj4*} moving at identical speeds each collide with a static hockey puck.
		Q: The puck hit by the [MASK] slides the {*shortest/longest*} distance.
Mass 2	720	**C:** A(n) {*mass_obj1*} and a(n) {*mass_obj2*} are placed on either end of a perfectly balanced seesaw.
		Q: The side of the seesaw with the [MASK] moves {*up/down*}.
		O: **A)** {*mass_obj1*} **B)** {*mass_obj2*} **C)** {*mass_obj3*} **D)** {*mass_obj4*}
Height 1	720	**C:** Four balls are dropped. The 1st is dropped from the height equivalent of a {*h_obj1*}, the 2nd is dropped from the height equivalent of a {*h_obj2*}, the 3rd is dropped from the height equivalent of a {*h_obj3*}, and the 4th is dropped from the height equivalent of a {*h_obj4*}.
		Q: The ball dropped from the height of the [MASK] takes the {*longest/shortest*} amount of time to fall.
Height 2	720	**C:** There are 4 staircases. The 1st staircase leads to the top of a {*h_obj1.*}, the 2nd staircase leads to the top of a {*h_obj2.*}, the 3rd staircase leads to the top of a {*h_obj3.*}, and the 4th staircase leads to the top of a {*h_obj4.*}.
		Q: The staircase leading to the top of the [MASK] is the easiest/hardest to walk up.
		O: **A)** {*h_obj1*} **B)** {*h_obj2*} **C)** {*h_obj3*} **D)** {*h_obj4*}
Circumf. 1	720	**C:** Four people are walking at identical speeds. The first walks around a {*circ_obj1*}, the second walks around a {*circ_obj2*}, the third walks around a {*circ_obj3*}, and the fourth walks around a {*circ_obj4*}.
		Q: The [MASK] takes the {*longest/shortest*} amount of time to walk around.
Circumf. 2	720	**C:** A person paints a circle around a {*circ_obj1*}, a {*circ_obj1*}, a {*circ_obj1*}, and a {*circ_obj1*}.
		Q: The circle around the [MASK] takes the {*most/least*} amount of paint.
		O: **A)** {*circ_obj1*} **B)** {*circ_obj2*} **C)** {*circ_obj3*} **D)** {*circ_obj4*}
Stackable	2400	**C:** A person is trying to stack {*stack*}, {*no stack1*}, {*no stack2*}, and {*no stack3*}.
		Q: The [MASK] are the {*easiest/hardest*} to stack.
		O: **A)** {*stack*} **B)** {*no stack1*} **C)** {*no stack2*} **D)** {*no stack3*}
Rollable	2400	**C:** A person is trying to roll a(n) {*roll*}, a(n) {*no roll1*}, a(n) {*no roll2*}, and a(n) {*no roll3*}.
		Q: The [MASK] is the {*easiest/hardest*} to roll.
		O: **A)** {*roll*} **B)** {*no roll1*} **C)** {*no roll2*} **D)** {*no roll3*}
Graspable	2400	**C:** A person is trying to move a pile of {*break*}, a pile of {*no break1*}, a pile of {*no break2*}, and a pile of {*no break3*} from one side of a room to the other using only one hand.
		Q: The [MASK] is the {*most/least*} likely to break.
		O: **A)** {*break*} **B)** {*no break1*} **C)** {*no break2*} **D)** {*no break3*}
Breakable	2400	**C:** A person drops a {*break*}, a {*no break1*}, a {*no break2*}, and a {*no break3*} from a balcony.
		Q: The [MASK] is the {*most/least*} likely to break.
		O: **A)** {*grasp*} **B)** {*no grasp1*} **C)** {*no grasp2*} **D)** {*no grasp3*}
Slideable	2400	**C:** A person is sliding four bricks across four hard surfaces. The 1st surface is covered with {*slide*}, the 2nd surface is covered with {*no slide1*}, the 3rd surface is covered with {*no slide2*}, and the 4th surface is covered with {*no slide3*}.
		Q: The surface covered with [MASK] is the {*hardest/easiest*} for the brick to slide across.
		O: **A)** {*slide*} **B)** {*no slide1*} **C)** {*no slide2*} **D)** {*no slide3*}
Bounceable	2400	**C:** A person is trying to bounce a rubber ball. They drop a first ball onto {*bounce*}, a second ball onto {*no bounce1*}, a third ball onto {*no bounce2*}, and a fourth ball onto {*no bounce3*}.
		Q: The ball dropped onto[MASK] bounces the {*most/fewest*} times.
		O: **A)** {*bounce*} **B)** {*no bounce1*} **C)** {*no bounce2*} **D)** {*no bounce3*}

Table 3.1: All templates in PROST. **C:** = Context, **Q:** = Question, **O:** = Options. {} indicate placeholders. The objects can be found in Table 3.3. Other placeholders show their possibilities in the braces themselves. [MASK] indicates the blank that the models need to fill. See Section 3.1.3 for more information.

NOTE: The number of objects with and without the affordances are swapped when the superlative is inverted.

Model Type	Input	Target
Decoder	*(context)* They are now walking $\langle \mathbf{O} \rangle$.	Max probability for each sentence input to the model
Encoder	*(context)* They are now walking $\langle \mathbf{M} \rangle$.	Max probability for the masked token.
T5	*(context)* They are now walking $\langle \mathbf{X} \rangle$.	$\langle \mathbf{X} \rangle$ south $\langle \mathbf{Y} \rangle$
UnifiedQA	Which way are they walking now? `\n` (A) north (B) south (C) east (D) west `\n` *(context)*	south

Table 3.2: Overview of the task preprocessing for different architectures evaluated. In all methods, the context remains unchanged and is "A person is walking west. They turn left."

3.1.3 PROST

PROST consists of $18,736$ cloze-style multiple-choice questions designed for probing a LM's physical reasoning ability. They cover 10 basic concepts: direction, mass, height, circumference, stackable, rollable, graspable, breakable, slideable, and bounceable. We choose these concepts because they are well defined, easily learned by interacting with the world, and are useful concepts for any embodied agent. The questions are constructed from 14 manually written templates. Each template follows one of three different formats: the first format is specific to the set of questions pertaining to directions; the second format is used to gauge the relative attributes—specifically mass, height, and circumference—of objects; and the third format targets the affordances of objects—specifically whether an object is stackable, rollable, graspable, or breakable, and whether a surfaces is slideable or bounceable[2] .

We use CheckList [310] to obtain the questions from our templates. We show all templates in Table 3.1 and explain them in detail below. We end this section by describing the objects featured in PROST.

Direction Templates We use two templates to generate questions which probe understanding of direction. The first focuses on cardinal directions. The second uses a set of four manually crafted questions to probe understanding of how gravity affects the directions of a ball throughout its trajectory. Due to their similarity, we count these four questions as a single template. The direction templates create a total of 16 questions.

---

[2] **Bounceable** here refers to providing an elastic collision.

Attribute Templates The second set of templates probe the models' ability to reason about relative mass, height, and circumference of common objects. For each of these three concepts we create a set of six objects that are easily ordered by their respective attributes. A context is first presented with up to four of the six objects to prime the models with the range of possible choices. This is followed by a prompt that probes the model to select one of the objects based on the object's mass, height, or circumference. By inverting the superlative in the prompt (e.g., *longest* ↔ *shortest*), we can probe the model's ability to identify both the object with the highest attribute value and the object with the lowest attribute value from the set of choices. We permute through all objects and all orders. Each of the three attributes are tested using two templates that share the same set of objects. Each template produces ${}_6P_4 * 2 = 720$ questions, meaning each attribute is probed using 1440 questions.

Affordance templates The remaining templates target an understanding of object affordances. For each affordance—stackable, rollable, graspable, breakable, slideable, and bounceable— we collect a set of five objects with and five objects without that affordance. Again, we first provide a short context that contains each of the four possible objects. We then provide a prompt that requires the model to select the only object either with or without the affordance. We include all permutations of objects where there is exactly one correct answer. These templates produce ${}_5P_1 * {}_5P_3 * 4 * 2 = 2400$ questions for each of the six affordances.

Objects in PROST All possible values for the placeholders in our templates are shown in Table 3.3. For affordances, we display objects in two groups: those with and without each affordance. For attributes, objects are sorted by increasing order, e.g., for mass, **leaf** is the lightest object and **microwave** is the heaviest object. Each object in PROST is selected to be single-token compatible for a wide range of vocabularies to enable easy probing of MLMs. We validate the order of our attribute objects and the group membership for our affordance objects by collecting judgments from 9 human validators. The validators obtained a 100% agreement on the object ordering, and 94.6% agreement on the object group membership.

Attributes	
Attribute	*Objects*
mass	leaf, coin, egg, apple, brick, microwave
height	book, microwave, table, car, house, mountain
circ	book, microwave, table, car, house, mountain

Affordances	
Affordance	*Objects*
stack	books, blocks, boxes, coins, plates
no stack	balls, bottles, eggs, flowers, lamps
roll	apple, ball, bottle, egg, can
no roll	book, box, block, mirror, microwave
grasp	balls, blocks, books, bottles, flowers
no grasp	flour, rice, salt, snow, sugar
break	bottle, egg, glass, mirror, plate
no break	ball, coin, pen, pillow, shirt
slide	ice, frost, grease, oil, soap
no slide	carpet, concrete, grass, gravel, rubber
bounce	asphalt, brick, concrete, rubber, steel
no bounce	carpet, foam, grass, leaves, snow

Table 3.3: Objects used in the templates.

3.1.4 Models

Using PROST, we probe three types of transformer-based models [381]: decoder models, encoder models, and encoder-decoder models. Each model has slightly different formatting requirements, which we show in Table 3.2. For each model type, we probe a range of different sizes to investigate the effects of scaling. We use Huggingface's [402] pretrained models, see Table 3.4 for the full set.

Model		Params (M)	Data (GB)
GPT		116.5	2
GPT-2	B	124.4	40
	M	354.8	40
	L	774.0	40
	XL	1557.6	40
BERT	B	110.1	13
	L	336.2	13
RoBERTa	B	124.7	160
	L	355.4	160
ALBERT V2	B	11.8	160
	L	17.8	160
	XL	59.0	160
	XXL	223.2	160
T5	S	60.5	170
	B	222.9	170
	L	737.7	170
	3B	2851.6	170

Table 3.4: Summary of models evaluated on PROST. We list the amount of pretraining data as the size of the uncompressed text corpus used.

Decoder Models We analyze OpenAI's GPT-1 [294] and GPT-2 [297]. Both are based on a transformer decoder architecture and trained on a traditional language modeling objective. We run these models over each question four times, each time with a different choice replacing the [MASK] token. Following [397], we select the sentence with the highest probability.

Encoder Models We analyze BERT (uncased) [84], RoBERTa [222], and ALBERT [196], which are all based on transformer encoders. BERT is trained on MLM and next sentence prediction and uses static masking, RoBERTa is trained on MLM with dynamic masking, and ALBERT uses whole-word n-gram masking. For probing, we filter out all but the four answer choices from the output vocabulary and select the token with the highest probability as the model's decision.

Encoder-decoder Models We also include results for T5 [301]. T5 is trained using a span corruption objective, in which spans of the input sequence are randomly replaced with a single mask token. During pretraining, span lengths are chosen randomly with an average length of three. To keep our results consistent with the other models, we restrict the span length to one token. We find that two of the options for sliding surfaces, namely **ice** and **frost**, violate our single-token constraint. To avoid any unfair comparison between answers that differ in token lengths and following previous work [123], we chose to omit presenting the results for T5 on the sliding concept.

Model		**Direction**	**Mass**	**Height**	**Circum.**	**Stack**	**Roll**	**Grasp**	**Break**	**Slide**	**Bounce**	**Macro Average**
GPT		46.7	40.1	24.3	22.8	28.2	27.9	19.6	22.7	14.6	14.4	26.1
GPT2		43.3	31.4	22.0	18.8	26.2	20.3	17.9	22.5	16.9	17.0	23.6
	M	48.3	34.1	21.6	21.6	25.5	23.7	24.9	27.8	22.5	18.5	26.8
	L	46.7	33.1	25.4	27.0	25.5	26.9	20.6	21.8	21.3	15.6	26.4
	XL	46.7	34.2	25.8	26.3	31.1	36.3	29.4	26.7	23.7	20.5	30.1
BERT	B	40.0	32.9	27.5	25.6	20.9	26.1	23.3	28.0	18.2	13.0	25.5
	L	**70.0**	38.8	19.4	17.5	21.3	19.2	26.7	19.5	15.9	18.6	26.7
RoBERTa	B	46.7	36.9	25.8	23.5	34.5	19.3	25.4	**45.0**	20.9	11.4	28.9
	L	66.7	**43.4**	33.8	22.7	22.7	22.2	29.4	23.8	22.7	25.5	31.3
ALBERT V2	B	21.7	35.4	30.2	26.0	25.2	32.5	35.3	22.8	15.3	22.9	26.7
	L	41.7	38.2	31.9	27.5	23.3	29.7	34.0	24.5	23.4	22.1	29.6
	XL	46.7	38.7	**42.0**	**37.4**	30.2	28.2	**37.1**	17.8	**25.3**	14.3	**31.8**
	XXL	68.3	33.8	28.1	24.5	29.4	23.4	21.2	30.2	17.5	22.1	29.8
T5	S	20.0	36.5	29.8	25.2	25.0	25.9	25.4	25.0	—	30.2	27.0^{*}
	B	40.0	37.0	32.6	23.8	25.0	23.4	25.2	25.6	—	**37.8**	30.1^{*}
	L	46.7	35.7	30.7	27.6	31.8	23.0	34.0	25.2	—	22.7	30.8^{*}
	3B	46.7	39.6	35.6	29.8	**34.7**	31.5	35.6	33.8	—	12.5	33.3^{*}
UnifiedQA	S	0.0	34.2	34.8	30.3	24.4	29.0	28.8	27.1	—	31.0	26.6^{*}
	B	0.0	17.8	33.3	22.3	25.5	34.9	27.9	36.5	—	45.7	27.1^{*}
	L	83.3	17.2	49.5	47.3	23.5	28.4	27.5	43.6	—	32.6	39.2^{*}
	3B	63.3	37.8	55.2	66.9	31.2	35.2	24.8	81.4	—	24.8	46.7^{*}
Task Average		46.3	36.5	28.6	25.2	27.1	25.9	27.4	26.0	19.9	19.9	28.5

Table 3.5: Macro average for each concept and overall for each model on PROST. The best accruacy for general pretrained-only models is displayed in bold. Note that the task average does not include UnifiedQA.

Finetuned Conditional LMs To better understand the limitations of text-only training, we additionally evaluate UnifiedQA [178]. UnifiedQA is a pretrained QA model, built off T5, and finetuned on SQuad 1.1, SQuaD 2.0, NarrativeQA, RACE, ARC, OpenBookQA, MCTest, and BoolQ [304, 303, 184, 193, 71, 249, 311, 69]. We format all of our templates to fit their multiple-choice question answering format and use their provided scoring metrics to select the models' answers.[3]

---

[3] https://github.com/allenai/unifiedqa

3.1.5 Results

The per model and per concept results are shown in Table 3.5. For concepts with more than one template—direction, mass, height, and circumference—we average across templates to get the concept's score.

We can see that, on average, ALBERT-V2-XL performs best, with an accuracy of 31.8%[4] , and GPT-2 performs worst, with an accuracy of 23.6%. We note that random guessing would yield an accuracy of 25%. Furthermore, every model underperforms random guessing on at least one concept. Since PROST is trivially solvable for humans, this supports our hypothesis that pretrained models are unable to perform physical reasoning anywhere close to human performance.

Comparing across all concepts, we see that direction obtains the highest average accuracy with 46.8%. The second best accuracy is observed for the mass attribute with 36.5%. The concepts models struggle the most with are the slideable and bounceable affordances, both with an average accuracy of 19.9%.

Model	**Position Accuracy**			
	1	**2**	**3**	**4**
GPT	27.0	24.3	7.6	38.6
GPT-2	29.9	23.1	8.1	42.0
BERT	28.4	24.3	5.7	38.2
RoBERTa	39.0	28.7	11.2	30.0
ALBERT V2	32.5	25.8	9.7	44.2
T5	52.4	21.1	1.9	35.2
UnifiedQA	41.0	27.7	18.8	51.9
Average	35.7	25.0	9.0	40.0

Table 3.6: Accuracy across the correct answer's position in the context.

---

[4] Note: as detailed in Section 3.1.4, T5 and UnifiedQA are not being evaluated on sliding. We therefore disregard their average accuracy.

Model		Mass	Height	Circum.	Stack	Roll	Grasp	Break	Slide	Bounce	Macro Average
GPT		9.4	2.2	14.0	35.9	43.0	22.7	13.3	9.8	10.1	17.8
GPT-2		19.2	24.3	1.1	16.1	5.1	12.1	31.9	24.9	15.8	16.7
	M	31.2	12.9	21.5	12.7	20.2	7.8	49.9	33.3	9.2	22.1
	L	25.8	24.2	25.4	5.6	16.7	18.4	24.6	28.2	23.7	21.4
	XL	43.5	6.7	1.5	56.1	51.5	36.3	31.5	15.8	32.4	30.6
BERT	B	5.0	40.0	2.5	12.3	15.4	3.1	44.9	12.2	11.2	16.3
	L	19.2	21.5	5.8	1.8	3.4	4.2	17.2	9.5	30.4	12.6
RoBERTa	B	4.7	4.0	6.5	55.0	13.8	27.8	89.6	21.8	15.8	26.5
	L	31.0	24.7	26.8	9.7	21.1	33.2	31.5	33.9	33.2	27.2
ALBERT V2	B	4.7	31.0	7.9	14.8	14.4	66.4	**2.3**	**7.2**	1.5	16.7
	L	**0.6**	11.9	23.6	30.0	36.8	52.2	6.9	28.7	13.2	22.7
	XL	9.4	**0.7**	8.5	19.5	8.1	31.0	10.9	8.2	20.6	13.0
	XXL	18.1	2.9	18.5	4.2	12.2	2.0	37.2	16.3	11.3	13.6
T5	S	8.3	12.9	13.5	**0.0**	3.9	0.8	3.1	—	1.8	**5.5**
	B	8.7	26.9	5.8	0.0	**0.2**	**0.5**	3.1	—	26.2	8.9
	L	5.0	20.3	1.7	7.5	22.6	10.5	37.7	—	**0.5**	13.2
	3B	16.1	12.2	**0.7**	9.2	8.8	5.1	34.7	—	24.9	14.0
UnifiedQA	S	46.9	5.4	19.7	2.8	34.5	31.0	39.0	—	9.3	23.6
	B	19.2	19.2	4.3	24.6	30.4	10.4	4.9	—	49.5	20.3
	L	18.5	28.5	26.2	18.6	28.1	41.6	7.9	—	48.1	27.2
	3B	8.2	46.2	36.1	6.8	1.8	0.7	13.4	—	26.4	17.5
Task Average		15.3	16.4	10.9	17.1	17.5	19.7	27.7	19.2	16.6	17.6

Table 3.7: Absolute difference in accuracy between a question and its superlative inverse.

3.1.6 Analysis

Object Order in Context For the concepts that use objects, all four choices are listed in each question's context. PROST contains all permutations with regards to their ordering. This enables us to directly look at the effect of the correct answer's position within the context on the models' accuracy. These results shown in Table 3.6.

We see that models have a strong tendency to select either the first or the last item seen in the context. The largest difference is found for T5, with an accuracy of 52.4% for objects at position 1 and an accuracy of only 1.9% for objects at position 3. We note that a proper understanding of the questions, as most humans would have, would be robust to the order in which the choices are presented. This further underlines that state-of-the-art models do not perform human-like physical reasoning.

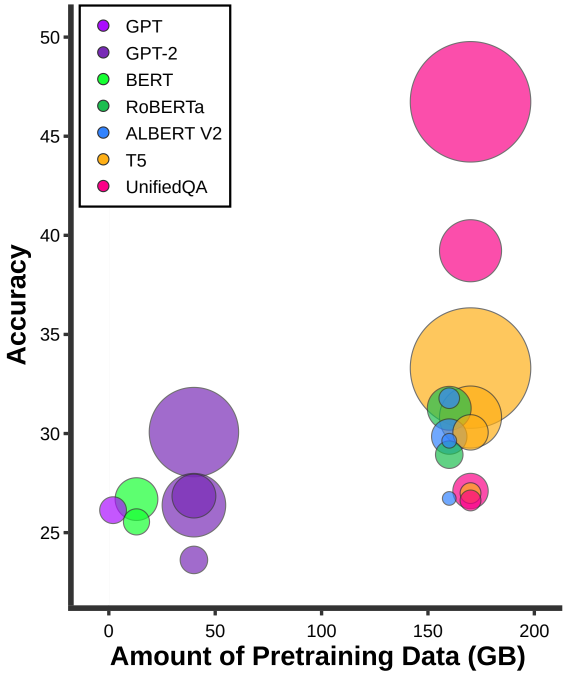


Figure 3.2: Scaling effect of models on accuracy. Circles size represents number of parameters.

Superlative Inverses By inverting the superlative in a question, we are able to probe a mirrored version of the question. For example, for attributes, this would require the model to identify the lightest object instead of the heaviest object, or, for affordances, it would require the model to identify the not stackable object instead of the stackable object. We call these mirrored versions **superlative inverses**. A true understanding of the questions in PROST should be robust to this kind of inversion. However, Table 3.7 shows all models perform better on one of the two versions. Of the probed models, GPT-2 is the most unbalanced, averaging 30.6% higher for one version over the other.

Data and Model Scaling Figure 3.2 shows each model's accuracy as a function of the number of its parameters. Unlike for many modern benchmarks, where increasing the number of parameters or training data provides significant benefits [363, 389], PROST does not see much improvement from such scaling. We observe some improvements with T5-3B outperforming T5-small, but this 6.6% increase requires a 48x increase in parameters and T5-small still outperforms T5-3B on one task. Moreover, some models break this trend: ALBERT's XL version outpeforms its XXL counterpart and GPT-2 M outperforms GPT-2 L. While previous work has revealed the impressive scaling laws of transformer-based architectures [171], PROST highlights the importance of relevant and informative training. As physical reasoning is not an ability that humans acquire via text, even substantially more open domain textual data is unlikely to lead to more than marginal improvements.

The Limits of Text-based Training To our knowledge, UnifiedQA is the most qualified model to succeed on our task, having been finetuned on a significant amount of relevant text data. While this additional data does provide benefits on PROST, it still falls short, with the best performing model we tested only achieving a 46.7% accuracy. Additionally, from Tables 3.6 and 3.7, it still lacks the robustness of proper understanding. This emphasizes that models are unlikely to obtain human-like reasoning from text-based training alone. Rather, PROST motivates exposing models to concepts through multiple modalities that mirror a human's experience.

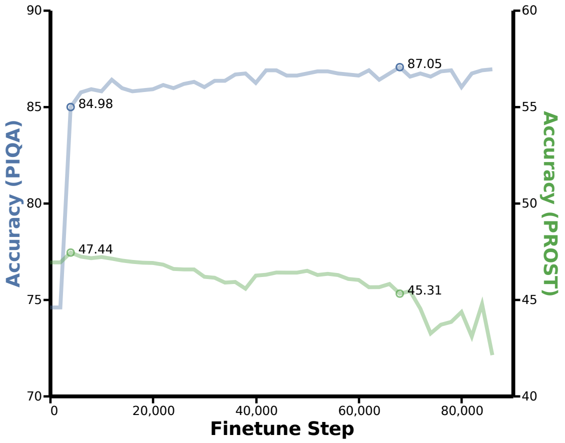


Figure 3.3: Analysis of the performance of UnifiedQA 3B on PROST throughout PIQA finetuning. The left and right Y axis represent Accuracy on the PIQA dev set and Macro accuracy on PROST respectively. We finetune for 100K steps, and compute metrics every 2k steps. Annotations correspond to the checkpoints with the best performance on PIQA and PROST. Note that PIQA has two answer choices, while PROST has 4.

Comparing PROST and PIQA Due to their shared focus on text-based physical reasoning, PROST and PIQA share similarities. To test if models trained on PIQA are able to carry over any concepts to PROST, we further finetune a UnifiedQA model on PIQA and evaluate it on PROST. The results, shown in Figure 3.3, indicate that training a model on PIQA is detrimental to its performance on PROST. While PIQA and PROST share a few conceptual similarities, they differ in terms of format, style, and vocabulary. We thus hypothesize that current models learn more about these surface-level differences than the conceptual similarities underpinning the questions. We further highlight two key differences between the two datasets:

- PROST probes models in a zero-shot fashion, whereas PIQA provides training and test sets of identically distributed examples. This makes it possible for models on PIQA to answer successfully using spurious correlations rather than physical reasoning.
- PIQA [35] covers an extensive range of objects and challenging physical concepts. [33] argues that experience is a prerequisite for understanding. It is hard to imagine how to expose a model to experiences ranging from egg yolk separation to making a pillow out of a garbage bag. In contrast, PROST provides a clear set of well defined concepts and objects that a model could potentially experience.

3.1.7 Discussion

Our experiments show that all the models we analysed fail to demonstrate a robust understanding of physical reasoning. Beyond performing poorly across every concept, they are easily influenced by changing the order of the objects in a question's context and by superlative inverses. Moreover, our analysis indicates that these issues are not likely to be solved simply by increasing the amount of model parameters or training data. All this evidence supports [28]'s and [33]'s theory that experience is a prerequisite of understanding.

A number of other reasoning benchmarks have been solved to some extent by a large finetuned model. UnifiedQA (11B parameters), based on T5 [301], achieved 81.4% on ARC [71]; and UNICORN [5] (11B parameters), also based on T5, achieved a 93.9% accuracy on hellaSWAG [415]. While all these models

---

[5] leaderboard.allenai.org/hellaswag/submissions/public

are larger and are trained on more data, our results force us to ask the question whether they perform well because these additional parameters and data have imbued the models with an ability to reason, or if they succeed by finding subtle unintended correlations in the data. This forces us to look more closely at how models succeed, and not just the accuracy they achieve. Tools like CheckList [310] can aid in this endeavor by demonstrating how robust models are to changes in the distribution of the data.

How to Use this Probe PROST is intended to help analyze any model that can be deployed in a text-only setting. However, we maintain that multi-modal data is necessary to experience the concepts in PROST, and that these experiences are likely a crucial step in succeeding on this dataset. One way that multi-modal models could prepare for this type of text-only evaluation is through multi-task training, where one of the tasks is only conditioned on text. Such an approach has already been considered: [49] propose an extension to their CLIP model which is trained on multiple modalities in a multi-task fashion. Because of the templated nature of PROST, its exact format can be adapted to match specific styles of language training, as we do for T5 and UnifiedQA.

PROST's language-only approach is motivated by two reasons. First, we believe that true multi-modal models should be able to function on any subset of their modalities. We note that humans can easily interact with text-only inputs (e.g., a text message) while still learning from and interacting with other modalities. Second, it enables the comparison of models trained using different modalities or domains. For example, we believe comparing how language understanding modules evolve when trained on vision-and-language navigation compared to visual question answering would provide invaluable insights.

Limitations We caution that achieving a high accuracy on PROST does not necessarily guarantee that a model is able of physical reasoning. It is likely easy to succeed on this benchmark if one were to intentionally train models on similar enough sentences or a subset of PROST itself. We hope that the community will use this dataset in the intended way: in a zero-shot setting to probe models which have been trained on data not specifically collected to succeed on PROST.

3.1.8 Conclusion

In this work, we introduce PROST, a probing dataset designed to evaluate a model's ability to reason about physical interactions. Our experiments reveal that state-of-the-art pretrained language models struggle with basic physical reasoning tasks, particularly when faced with subtle linguistic perturbations such as reordered options or inverted questions, challenges that do not typically confuse humans. These findings underscore a critical gap in current LLM training: a lack of grounded, embodied experience. Moreover, our analysis suggests that this gap cannot be bridged through model scaling alone. These results reinforce the broader need to move beyond text-only data and to incorporate the kinds of rich, structured data sources that support human-like generalization.

3.2 The World of an Octopus: How Reporting Bias Influences a Language Model's Perception of Color

The work described in this section has been published in EMNLP 2021 [271].

3.2.1 Introduction

Given sufficient scale, language models (LMs)[6] are able to function as knowledge bases, yielding factoids and relational knowledge across a wide range of topics [283, 45]. However, subsequent work [28, 33, 15] has raised concerns about the inherent limitations of text-only pretraining. Motivated by these concerns and limitations, we identify and investigate how reporting bias, a concrete and measurable signal, correlates with these limitations and how multimodal training can mitigate these issues.

Grice's conversational maxim of quantity [257] asserts that utterances only contain the required amount of information. This leads to explicit reporting of self-evident knowledge being rare, while less common facts, properties, or events are being reported at disproportionately high frequencies. For example, while most people agree that bananas are typically yellow, the bi-gram "green banana" is 332% more frequent in the Google Books Ngram Corpus [216] than "yellow banana".[7] This reporting bias inevitably propagates

---

[6] In this paper, we use LM to refer to both causal LMs as well as masked LMs.

[7] We calculate this number using version 3 from February 2020.

from corpora to the models trained on them [339] and affects a variety of concepts. One such concept that we would expect to be harmful in downstream applications, is easy to measure, and is solvable via visual input is color. For these reasons, we investigate the relationship between reporting bias and modern LMs' perception of color.

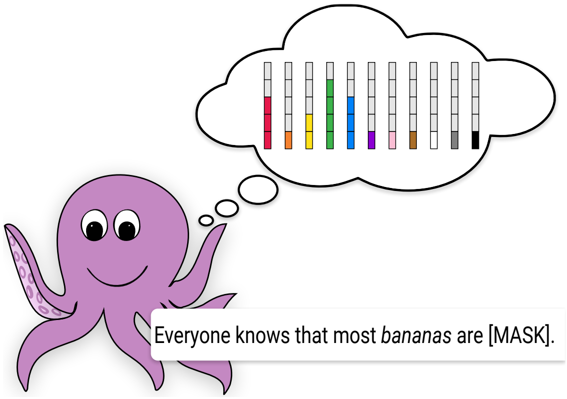


Figure 3.4: An example prompt from CoDa.

People's understanding of color is primarily derived from their experience in the world. Every time we interact with an object, we update our understanding of the possible colors that object can take on. Further, we can often apply meaning to the differences: a green banana is unripe, a yellow banana is ideal, and a brown banana may be past its prime. Text-only LMs do not share this embodied experience. Similar to an octopus[8] they cannot see colors, and need to rely solely on the inaccurate reporting of colors in text. Thus, we expect the colors LMs associate with objects to differ drastically from a human's perception.

To test this hypothesis, we construct the Color Dataset (CoDa) – a ground-truth dataset of color distributions for 521 well-known objects via crowd-sourcing. We use this dataset to compare the color distributions found in text and those predicted from LMs, finding that a LM's shortcomings in recovering color distributions correlates with the reporting bias for those objects. Next, we hypothesize that models having access to multiple modalities, specifically vision and text, may be able to partially overcome these shortcoming by grounding the language to their limited visual experiences [33]. To this end, we develop a unified framework for evaluating the color perception of text-only and multimodal architectures. Our results

---

[8] The octopus is a species which has no color photo-receptors and is the protagonist of the thought experiment in [28].

support the hypothesis that multimodal training can mitigate the effect of reporting bias.

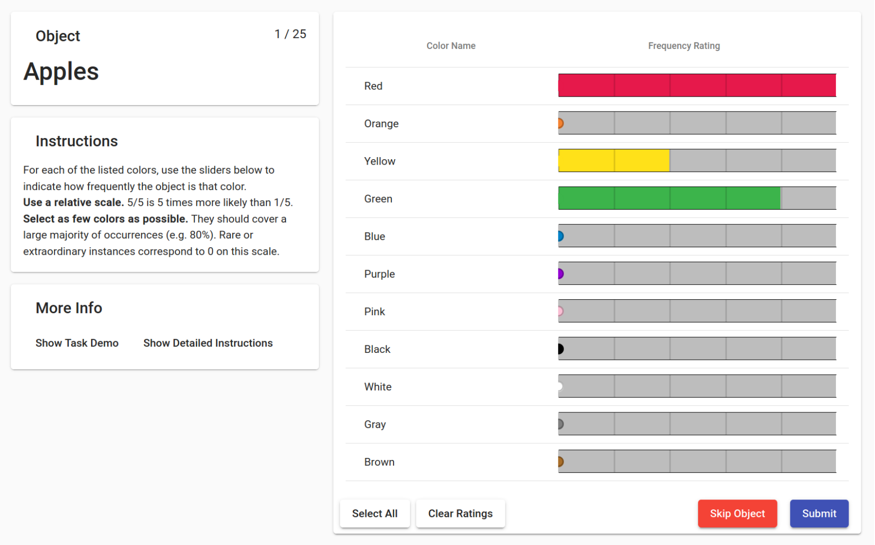
Contributions We make three contributions: 1) We introduce a dataset with human color distributions for 521 well-known objects. 2) We conduct an extensive analysis to identify how reporting bias affects LMs' perception of color. 3) We demonstrate that multimodal training mitigates, but not eliminates, the impact of reporting bias.

3.2.2 CoDa

3.2.2.1 Dataset Creation

Dataset	**Count (Percentile)**		
	25%	**50%**	**75%**
Open Images V6	2.10K	3.96K	11.1K
Google Ngrams	1.63M	4.64M	25.4M
Wikipedia	2.04K	10.3K	38.8K
VQA	4	25	186

Table 3.8: Object frequencies in each domain/dataset after filtering. We report class label statistics for Open Images and n-gram frequencies for Google Ngrams, Wikipedia, and VQA prompts.



Object 1 / 25

Apples

Instructions

For each of the listed colors, use the sliders below to indicate how frequently the object is that color.
Use a relative scale. 5/5 is 5 times more likely than 1/5.
Select as few colors as possible. They should cover a large majority of occurrences (e.g. 80%). Rare or extraordinary instances correspond to 0 on this scale.

Figure 3.5: Our task UI for data collection on Amazon Mechanical Turk. See Section 3.2.2.1 for full details.

Object Selection To ensure all our models – and potential future models – are properly exposed to the objects in our probing dataset, we choose objects which are common in both text and image data. We start with objects from the Open Images dataset [192] and remove all objects which appear less than 25 times in Wikipedia. For example, we remove "dog bed" as the corresponding bi-gram only appears 19 times. This leaves us with an initial set of 687 objects.

We then manually filter out all human-related words, such as "person" as well as hypernyms such as "food", since they are too general to assign specific colors. We also remove transparent objects, such as "windows", and objects that are more than two words long, such as "personal flotation device" and "table tennis racket". This leaves us with our final set of 521 objects. We provide object frequencies from Open Images V6 [192], the Google Books Ngram Corpus [216], Wikipedia, and VQA [131] in Table 3.8.

Color Selection Following [31], we choose the 11 basic color terms of the English language as the colors to be annotated: red, orange, yellow, green, blue, purple, pink, black, white, grey, and brown.

Color Annotation Due to sample bias in image datasets [374] and the difficulty of matching pixel values to human perception, generating color distributions by counting color frequencies in images is impractical and challenging to verify.[9] Thus, in line with our focus on human perception of color as it relates

---

[9] We attempted an image search paradigm, but challenges such as varied lighting, imperfect segmentation, and the complexity of aligning colors to human perception meant that such a method would still have required human verification.

to language (i.e., color terms), we approximate color distributions via human annotation crowd-sourced on Amazon Mechanical Turk (MTurk).[10]

Workers are shown words representing objects and tasked with rating – on a scale from 1 to 5 – the frequency with which instances of the objects appear in each of the 11 provided colors. We set up these tasks as human intelligence tasks (HITs), and provide the workers with instructions, which include an example for how one could label "grass" and a concrete list of acceptance and rejection criteria. Each HIT includes 25 objects and is compensated with $1. Figure 3.5 shows the user interface as presented to an MTurk worker tasked with annotating the object "apples".

Since we choose objects that appear frequently in datasets, we expect people to be familiar with them. However, for the rare cases where an annotator is unsure about an object's color, our interface includes a skip button. The average crowd-worker skips 1 object. If an object is not skipped, the average worker completes one annotation in 14 seconds on average. Each object's annotation is normalized to obtain a probability distribution over colors.

A potential side-effect of crowd-sourcing annotations is that annotators might choose fewer colors to minimize the time spent on the task. In light of this, we design a labeling interface that balances the time required for labeling a given object as one, many, or all colors. For example, we include a "Select All" button and use wide click-optimized sliders. With these changes, we find that, on average, users tend to select 6.2 colors per object. For more details and analysis regarding annotator biases, we refer the reader to Appendix B.2.

Quality Control For quality control purposes, each HIT includes "spinach" as a control object at a random position within the group of objects to annotate. This control object serves as a way to flag any submissions which do not follow the instructions or are otherwise not suitable for our purposes.[11] We require the rating of "spinach" to be more than 50% green in order to accept the HIT. Rejected HITs are not included in the dataset. This filters out the small number of workers who provide random or blatantly incorrect annotations.

---

[10] This project went through our institution's ethics review before crowd-sourcing was initiated.

[11] Annotators are made aware that control objects with known color distributions are included in the HIT.

We compute the ground truth as an average over all submitted annotations for a given object. We iteratively filter annotations on a per-object basis if a rating has a Kendall correlation of less than 0 with the current ground truth. This removes 10 annotations that appear to be cases of annotator misinterpretation. For example, one annotator labels “stop sign” as being equally red, yellow, and green, likely confusing “stop sign” with “traffic light”.

Group	**All**	**Train**	**Val**	**Test**	**Examples**
Single	198	118	39	41	Carrot, Spinach
Multi	208	124	41	43	Apple, Street light
Any	115	69	23	23	Shirt, Car
Total	521	311	103	107	

Table 3.9: CoDa splits by object group.

Object Grouping We are investigating the relationship between LMs’ knowledge of object colors and reporting bias, the tendency of humans to not state the obvious [257]. We hypothesize that reporting bias will be more severe for objects which have a single typical color, as that color will be implicitly assumed by a listener or reader and, accordingly, will be less frequently stated explicitly. In contrast, objects with a distinct set of several possible colors require explicit descriptions to fully capture the visual characteristics of the object. For example, apples are often described as red or green.

To test whether objects with different color distributions are impacted by reporting bias differently, we divide the dataset into three categories: single-color objects, multi-color objects, and any-color objects. We categorize objects using k-means clustering with the Jensen-Shannon distance of sorted probabilities. This creates clusters which are color-invariant and based only on the properties of the distributions. We find that this method gives consistent clusters, i.e. the clusters are independent of seeding. We then assign group names semi-manually.[12] “Lemon” is an example of a single color object, where 73% of the distribution is yellow. “Wine” is a multi-color object with 90% of the distribution falling on red, white, pink, and purple (the last 10% is yellow). All other objects are any-color objects: they have no clear set of typical colors.

---

[12] As there are 3 groups, we can simply mark the “extreme” clusters as Single and Any.

Examples of any-color objects are t-shirts, cars, or flowers. More examples are shown in Table 3.9.

Model Type	**Input**
Decoder	Most apples are $\langle \mathbf{O} \rangle$.
Encoder	Most apples are $\langle \mathbf{M} \rangle$.
CLIP	A photo of an apple.

Table 3.10: Example inputs for different evaluated architectures.

3.2.2.2 Templates

Text-only corpora and visually-grounded datasets rarely occupy the same domain. To accommodate both, we form a set of templates for each domain. The first is tailored to text-only models, and consists of both plural templates such as “Most bananas are [MASK].” and singular templates such as “This banana is [MASK]”.

Our second template group is tailored to visually-grounded datasets. We use most of the templates provided by [295], which the authors used for finetuning on ImageNet, but exclude templates that inherently point to an unnatural object state, such as “a photo of a dirty banana”. Examples for templates are provided in Table 3.10.

We recognize that any hand-crafted templates are by nature imperfect. As such, we use all configurations for all models and present the best results per-object for each model to give models ample opportunity to succeed.

3.2.2.3 Data Splits

Some of our experiments (cf. Section 3.2.4.2) require a small training set. Thus, CoDa contains training, development and test splits, with 311, 103, and 106 objects respectively. There is no object overlap between the different sets.

Dataset	**Group**	**Freq**	**Spearman $\rho \uparrow$**	**Kendall's $\tau \uparrow$**	**Acc@1 $\uparrow$**	**$\mathbf{D_{JS}} \downarrow$**
Google Ngrams	Single	5.60	41.7 ± 27.8	35.3 ± 24.5	**43.9**	0.27 ± 0.16
	Multi	9.69	$\mathbf{47.1} \pm \mathbf{26.6}$	$\mathbf{38.1} \pm \mathbf{22.2}$	30.3	0.23 ± 0.12
	Any	20.26	43.5 ± 30.7	34.3 ± 25.0	33.9	$\mathbf{0.15} \pm \mathbf{0.10}$
Wikipedia	Single	1.51	26.5 ± 30.2	22.2 ± 26.3	**25.3**	0.37 ± 0.17
	Multi	1.85	29.4 ± 31.9	$\mathbf{23.9} \pm \mathbf{27.0}$	23.6	0.31 ± 0.16
	Any	3.00	$\mathbf{30.9} \pm \mathbf{31.5}$	23.8 ± 25.6	19.1	$\mathbf{0.23} \pm \mathbf{0.15}$
VQA	Single	0.73	27.4 ± 37.8	25.4 ± 35.3	16.7	0.38 ± 0.23
	Multi	2.17	$\mathbf{35.7} \pm \mathbf{34.3}$	$\mathbf{31.7} \pm \mathbf{30.9}$	21.2	0.35 ± 0.20
	Any	2.64	33.7 ± 33.6	28.1 ± 28.7	**27.8**	$\mathbf{0.29} \pm \mathbf{0.17}$

Table 3.11: Correlation metrics between the n-gram frequencies reported in different datasets and the ground truth distributions collected from human annotators. Single, Multi, and Any indicate sets of objects that are frequently a single color, between two to four colors, or could be any color, respectively. We aggregate by object and report the mean ± standard deviation for each metric across the objects of that group.

3.2.3 Reporting Bias

3.2.3.1 Background

As previously stated, Grice's conversational maxim of quantity manifests as *reporting bias* – i.e., people not usually stating obvious facts or properties –, and impacts nearly all datasets that contain text.

Reporting bias has been studied in the context of both NLP and image captioning. [130] perform a quantitative analysis using n-gram frequencies from text, finding this phenomenon particularly relevant to internet text corpora. [339] extend these experiments to pretrained models such as Bert [84] and RoBERTa [222]. Similar to our work, they analyze color attribution of the form "The _____ banana is tasty." However, their ground truth is extracted from Wikipedia bi-grams and, thus, suffers from reporting bias itself. In contrast, we circumvent this problem by collecting the ground truth in CoDa directly from humans.

3.2.3.2 Reporting Bias in Text

Our hypothesis is that pretrained LMs inherit reporting bias with respect to colors from their training data. Thus, prior to our main experiments, we investigate if, in fact, reporting bias exists in large general text corpora. We analyze the Google Books Ngram Corpus [216] and Wikipedia. Specifically, we look at all bi-grams and tri-grams containing a color followed by an object in our dataset.

Let us denote the count of the n-gram $x_1 \ldots x_n$ as $\phi(x_1, \ldots, x_n)$. We then define the relative frequency with which each object o appears with a color c as:

$$\text{Freq}(o) = \frac{100}{\phi(o)} \sum_{c \in C} \phi(c, o) \tag{3.1}$$

We further define the probability of an object being of color c^* as:

$$P(c^* \mid o) = \frac{\phi(c^*, o)}{\sum_{c \in C} \phi(c, o)} \tag{3.2}$$

The results of these experiments are reported in Table 3.11. The frequency column supports our hypothesis that objects with one typical color are less frequently described as being of any color than those with multiple typical colors or where any color is possible. In all metrics excluding Acc@1, the text-retrieved color distributions are more strongly correlated with the ground truth for multi and any colored objects than for single-colored objects.[13]

3.2.4 Experimental Setup

Model	**Sizes**	**Multimodal**
GPT-2	B, M, L, XL	
RoBERTa	B, L	
ALBERT V1	B, L, XL, XXL	
ALBERT V2	B, L, XL, XXL	
CLIP	ViT-B/32, RN50, RN50x4, RN101	✓

Table 3.12: Summary of evaluated models.

---

[13] Acc@1 is not directly comparable across object groups, see Section 3.2.4.4 for details.

3.2.4.1 Zero-shot Probes

We first probe LMs in a zero-shot fashion using a set of templates (see Section 3.2.2.2). Each template has a [MASK] where the color should appear. For models trained using a causal language modeling objective, we run the models over each template eleven times, each time with a different color replacing the [MASK] token. Following [397], we select the sentence with the highest probability. For models trained using a masked language modeling objective, we filter the output vocabulary to only include the eleven color choices and normalize to obtain a probability distribution.

3.2.4.2 Representation Probes

Many current multimodal architectures are optimized for multimodal evaluation and have complex shared embedding spaces, which makes it challenging to compare to text-only models. However, recent developments such as CLIP [295] and ALIGN [167] show promising results in connecting images and text via contrastive pretraining on large unlabeled corpora, while still maintaining separate text and image models. We focus on probing multimodal models which follow these architecture decisions. Since they have not been trained on a language modeling objective, zero-shot probing is not viable on these models. To overcome this and enable comparison to text-only models, we freeze the base model and use part of our dataset to train a MLP to extract color distributions from the frozen representations.

Given pretrained representations, we would like the performance of a model to consist of 2 parts: final quality (in our case distribution correlations), and the amount of effort to get that quality from the representations. This is possible by formulating the task as *efficiently* learning a model from representations to color distributions. Following [400, 386], we conduct our experiments for representation probing in a loss-data framework using minimum description length (MDL), surplus description length (SDL), and ε sample complexity (εSC). We split the training set into 10 subsets spaced logarithmically from 1 to 311 objects, and report averages over 5 seeds.

Model	**Group**	**Spearman $\rho\uparrow$**	**Kendall's $\tau\uparrow$**	**Acc@1$\uparrow$**	**$\mathbf{D_{JS}}\downarrow$**	$\Delta\rho\uparrow$	$\Delta\tau\uparrow$
GPT-2	Single	40.3 ± 26.6	33.6 ± 22.1	**40.4**	0.39 ± 0.07	-0.55	-1.01
	Multi	44.8 ± 20.9	36.5 ± 16.8	29.8	0.26 ± 0.06	-1.49	-1.05
	Any	**48.1** ± **25.1**	**38.2** ± **20.2**	40.0	**0.09** ± **0.04**	**5.29**	**4.46**
RoBERTa	Single	47.8 ± 24.7	40.1 ± 20.8	**42.9**	0.28 ± 0.11	7.17	5.69
	Multi	50.2 ± 23.8	41.0 ± 19.5	33.2	0.19 ± 0.08	4.57	4.01
	Any	**52.5** ± **23.5**	**42.0** ± **19.5**	36.5	**0.10** ± **0.06**	**9.97**	**8.26**
ALBERT	Single	43.7 ± 24.4	36.4 ± 20.6	34.3	0.30 ± 0.11	2.69	1.55
	Multi	44.6 ± 19.1	36.1 ± 15.5	26.9	0.22 ± 0.07	-1.53	-1.27
	Any	**48.2** ± **21.4**	**38.2** ± **17.2**	**35.7**	**0.11** ± **0.05**	**5.07**	**4.22**

Table 3.13: LM results when probed in a zero-shot setting. Single, Multi, and Any indicate sets of objects that are frequently of a single color, between two to four colors, or could be any color, respectively. All correlation coefficients (ρ, τ) are multiplied by 100. For each object, we take the prediction from the template with the highest τ correlation. We then aggregate by object and report the mean ± standard deviation over objects of that group. We report the results from the best model from each architecture; for results on a per-model basis, see Table B.1.

3.2.4.3 Models

We probe object-color probabilities in 14 pretrained text-only models as well as four versions of CLIP [295]. The text-only models are varied configurations of GPT-2 [297], RoBERTa [222], and ALBERT [196]; cf. Table 3.12 for the full set. We use Huggingface's [402] pretrained models for all text-only models and the official implementation of CLIP.[14]

3.2.4.4 Metrics

In order to obtain as comprehensive a picture as possible, we report a variety of metrics when applicable, including: top-1 accuracy, Spearman rank order correlation ρ, Kendall rank correlation τ, and Jensen-Shannon divergence D_{JS} for each model and each set of objects. Each of these metrics highlight slightly different aspects of performance on the task.

---

[14] github.com/openai/CLIP

Top-1 accuracy (Acc@1) is the frequency with which models can correctly identify the most frequent color of an object. This is useful for comparing models, but not directly interpretable across object groups as it inherently favors objects that can take on few colors. Spearman's ρ is sensitive to outliers, so it highlights the extreme mistakes, while Kendall's τ is more robust to such changes. Jensen-Shannon divergence measures the similarity between 2 distributions.

Spearman's ρ and Kendall's τ are within the range of $[-1, 1]$, with -1 being negatively correlated and 1 being perfectly correlated.[15] We additionally define $\Delta\rho$ and $\Delta\tau$ correlation difference measures defined on the interval [-100, 100], to compare model predictions to n-gram frequency predictions. This measures the difference in correlation between n-gram frequency predictions and a model's probability distribution, where -100 indicates degraded correlation, 0 equals perfect correlation, and 100 indicates improved correlation with the ground truth as compared to the relative n-gram frequencies. In the context of reporting bias, $\Delta\rho$ and $\Delta\tau$ can be interpreted as measures of bias amplification or mitigation for negative and positive values, respectively.

We additionally define an average of the two correlation metrics as "Avg. Correlation". When using this metric, we first compute $\frac{\rho+\tau}{2}$ for a specific object and perform all other aggregations in the same way as for the other metrics.

---

[15] We multiply by 100 in all tables for legibility.

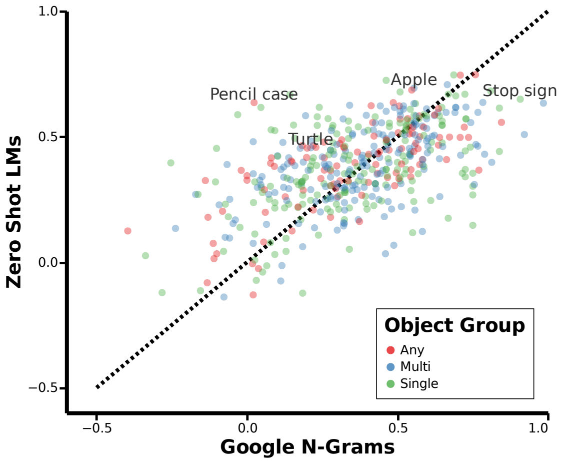


Figure 3.6: Correlation between n-gram frequency and LM performance for single, multi, and any color objects. X and Y axes are Kendall's τ correlation between n-gram frequency and ground truth and LM predictions with ground truth respectively. Each point corresponds to a single object in our dataset. LM correlation is averaged over the top models for each architecture. The dotted line y=x corresponds to to perfect correlation.

3.2.5 Results

3.2.5.1 Zero-Shot Probes

The results of LMs when probed in a zero-shot setting, provided in Table 3.13, clearly demonstrate that LMs perform worse on single-color objects and perform better on objects that can take on a range of colors. Furthermore, correlations are relatively low for all objects and models. This demonstrates that colors are generally challenging for state-of-the-art pretrained LMs.

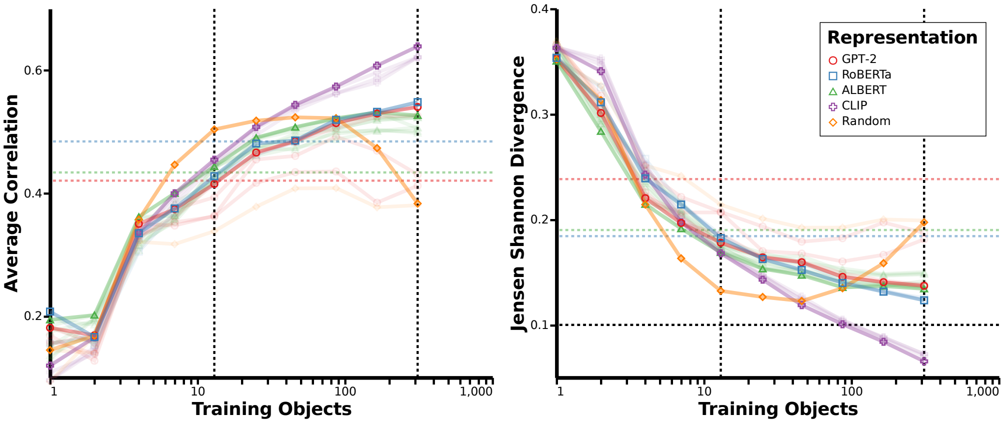


Figure 3.7: Representation probing results for unseen objects with varying amounts of data, averaged over 5 seeds. The main lines are the best model from those of the same type (e.g., RoBERTa$_{\text{BASE}}$ and RoBERTa$_{\text{LARGE}}$), and the translucent lines are the per-model averages. Dotted lines represent best zero-shot performance for each model. The "Random" group consists of a randomly initialized RoBERTa and CLIP. The black dotted lines correspond to ϵ and n in Table 3.15. Left: Average of Spearman's ρ and Kendall's τ. Right: Jensen-Shannon divergence.

3.2.5.2 Reporting Bias and Model Accuracy

Figure 3.6 compares the correlation between n-gram frequency and zero-shot LM performance. The identity line represents a theoretical perfect correlation between how well n-gram frequency correlates with our ground truth and LM predictions.[16] Any points above the identity line represent cases where LMs seem to *mitigate* reporting bias – their predictions are closer to ground truth, and points below the line represent cases where LMs *amplify* reporting bias – their predictions are further from ground truth. When averaged across all models (see Appendix B.4 for the full list of results) zero-shot LMs amplify the reporting bias of single-color objects by 5.23% on average, and 6.26% for multi-color objects. For any-color objects, we find a slight mitigation of 0.21% on average.

Table 3.14 aggregates and combines results from Tables 3.11 and 3.13 and elucidates two main points on the effect of reporting bias on a LM's perception of color. First, the color distributions of LMs correlate more strongly with reporting bias-affected text than with a human's perception of color. Second, single-

---

[16] That is, where LMs directly reflect n-gram frequencies.

colored objects are the most affected by reporting bias, and the objects LMs struggle the most on. These results indicate that, in line with our hypothesis, LMs are negatively impacted by reporting bias. Further, because reporting bias is innate to human communication and due to the enormous amount of text required for modern LMs, it is infeasible to eliminate reporting bias from all training data. This entails – in support of the arguments in [28] and [33] – that language understanding abilities are naturally limited by text-only training.

		Avg. Correlation ↑	
Group	**Freq.**	**Humans**	**Ngrams**
Single	5.60	40.1 ± 22.3	63.0 ± 18.1
Multi	9.69	42.2 ± 20.5	63.1 ± 17.5
Any	20.26	42.9 ± 22.5	$\mathbf{63.4} \pm \mathbf{16.2}$

Table 3.14: **LM predictions have higher correlation with n-gram frequencies**. Here we compare the average correlation between LM predictions and two sources of "ground truth"; one collected from human annotators and one computed from n-gram frequencies. Single, Multi, and Any indicate sets of objects that are frequently of a single color, between two to four colors, or could be any color, respectively. The "Freq." column indicates the frequency n-grams containing these objects also have one of the eleven colors.

		GPT-2	**RoBERTa**	**CLIP**
n		L	B	ViT-B/32
13	$\mathbf{D_{JS}}$	0.178	0.185	0.168
	MDL	2.80	2.95	2.79
	SDL, ε=0.1	$>$ 1.50	$>$ 1.65	$>$ 1.49
	εSC, ε=0.1	$>$ 13	$>$ 13	$>$ 13
	Avg Corr.	40.7	42.7	45.5
311	$\mathbf{D_{JS}}$	0.137	0.123	**0.065**
	MDL	45.07	42.08	**27.22**
	SDL, ε=0.1	$>$ 13.97	$>$ 10.98	**2.43**
	εSC, ε=0.1	$>$ 311	$>$ 311	**165**
	Avg Corr.	54.0	54.9	**63.9**

Table 3.15: Estimated measures of representation quality for the best model of each architecture.

3.2.5.3 Representation Probes

Figure 3.7 shows the average correlation and Jensen-Shannon divergence for unseen objects as a function of the number of training objects. Note that with 14 objects, all models surpass zero-shot performance in terms of Jensen-Shannon divergence. With enough training objects, we observe similar ranking patterns observed in the zero-shot setting for text-only models. However, the advantage of this approach is that we can additionally include multimodal architectures.

The results from these experiments demonstrate that multimodal models outperform text-only models at recovering color distributions. They manage to do so even though the performance of multimodal models is often lower on classic NLP tasks [365] and many multimodal datasets are even more prone to reporting bias in text [255, 380, 52]. This further support the arguments in [33] that understanding concepts requires experiencing them in their natural form.

3.2.6 Limitations

While our work identifies issues with text-only training and motivates the use of multimodal signals during pretraining, in this section we outline some limitations of our approach.

First, a number of recent papers have highlighted potential limitations of probing LMs in certain ways

[418, 400]. While we acknowledge that probing does not provide a full picture of the capabilities of LMs, our hypothesis was supported by a range of different results from different approaches. In future work, we hope to leverage research [45, 168] that demonstrates effective methods for automatically producing templates optimized for specific models. In the current state, we cannot and do not state exactly what LMs do and do not capture, rather we use our results to uphold and strengthen our original hypothesis that reporting bias hinders performance and that multimodal signals can help mitigate this problem.

Second, the bi-gram/tri-gram approach we use to quantify reporting bias only approximates the full set of object-color instances. To be more exact, a dependency parser would have to be run on every dataset.

Finally, although our results motivate the use of multimodal signals during pretraining, there are still challenges to overcome. As discussed by [365], the performance of multimodal models on classic NLP tasks often does not reflect the inherent advantages of these architectures, and many multimodal dataset are even more prone to reporting bias in text [255, 380, 52]. Further, while a visual signal is able to better impart a sense of color, it is not enough to endow models with the meaning behind those colors. Humans easily learn that a green banana is not yet ripe, and that a brown banana is past its prime. For models to obtain this level of knowledge and reasoning they will likely require training signals from more modalities, and potentially fully embodied experiences.

3.2.7 Related Work

Color-Object Relationships Preexisting word association datasets often include object-color relationships as either having multiple equally likely pairings [119, 191], or as probabilistic cue-target pairs [258]. Others such as [83] take a norm completion approach, wherein participants are tasked with generating attributes given some concept. One can then extract the object-color relationships by counting the number of participants who reported a given color.

However, the resulting “distribution” is an aggregate count over individuals, and does not necessarily reflect the distribution from the eyes of a single observer. Thus, previous research into LMs as knowledge bases has not been able to fully explore the extent to which they know color [1, 339].

Previous work has shown the importance of color in visual perception and object recognition [317, 112].

More recently [371] use time resolved neural imaging data to demonstrate how the typicality of object-color relationships influences object representations in visual processing.

Probing LMs A wide range of papers have probed LMs in a zero-shot fashion by looking at how they fill in a [MASK] token in handcrafted [399, 283, 168, 97, 212] or automatically generated [45, 168] template sentences. Others, such as [397] compare perplexities between minimal pairs of sentences. A different approach is to analyze the representation quality of LMs for linguistic tasks by training a simple MLP on pretrained model representations [79, 215]. However, [418] demonstrate that the procedure of training an additional classifier may distort the results. An alternative approach introduced by [386] is information-theoretic probing with MDL. This method builds on standard probing classifiers by not only measuring the final performance, but additionally measuring the amount of effort required to achieve that performance.

Probing Multimodal LMs Often multimodal LMs are used in the domain of visual question answering, where, given an image, the model is asked a question about concepts in the image [131, 158]. While it is often possible to simply use the text-only portion of these models for other tasks, this often leads to poor performance on solely language-based tasks [365].

3.2.8 Conclusion

In this work, we investigate how reporting bias—the human tendency to omit commonly understood information—can distort a language model's perception of color. To study this, we introduce CoDa, a dataset capturing human-perceived color distributions for 521 common objects. Using CoDa, we show that text-only models consistently underperform due to gaps introduced by reporting bias, revealing a fundamental limitation in language modeling that stems from relying solely on text data. We further demonstrate that multi-modal training, which incorporates perceptual signals, effectively mitigates these issues. These findings further support the broader argument that effective language understanding requires access to grounded, non-textual experience, and reinforce the need to explore multimodal approaches as a path toward more generalizable AI systems.

3.3 ReSeeding Latent States for Sequential Language Understanding

The work described in this section is under review for EMNLP 2025 [11].

3.3.1 Introduction

The continued scaling of large language models (LLMs) has led to impressive capabilities across a range of natural language tasks. Yet, the field is nearing the limits of available high-quality text data [385], and models trained solely on text data exhibit persistent limitations in their compositional reasoning [91], planning [379], and length generalization [407]. These challenges motivate the integration of non-text modalities—often referred to as *grounding*—to enhance model capabilities. However, existing grounding approaches either explicitly depend on auxiliary modules at inference, or implicitly align encoder-only models that lack generative capacity. We introduce Refeeding State Embeddings aligned using Environmental Data (RESEED), a flexible framework to directly ground decoder-based LLMs in structured environment data, leveraging both implicit and explicit signals. We show that RESEED improves sample efficiency, length generalization, and compositional reasoning in long-horizon sequential tasks.

Modern LLMs are trained in three stages: unsupervised causal language modeling (CLM) [294], a supervised finetuning with CLM, and alignment via preference optimization [269]. Throughout this process, models are exposed to *text data* and *human preference data*. While both have been instrumental to recent progress, they omit key elements required for human-like language understanding. Text offers linguistic structure and encodes world knowledge [34], but abstracts away key spatial, temporal, and causal relationships between concepts [28]. Moreover, text tends to omit self-evident information, resulting in reporting bias that negatively impacts language modeling [257, 271]. Human preference data, while useful for alignment, is both sparse—a single bit for sequences of text—and subjective—some humans may prefer a more grammatical output, while others a more factual output. *To this end, we posit that a third kind of data is required:* ***data from the environment***. Motivated by research outlining the necessary role a human's interaction with their environment plays in language understanding [120, 108], we hypothesize that structured environmental signals can improve language modeling. Environment data, which we define as sequences of states that capture how an environment changes, complements text and human preference data in four key ways: (1) it preserves spatial and temporal relations; (2) it is concrete and fully specified, avoiding abstraction and reporting bias; (3) it provides a dense and informative training signal; and (4) it is consistent and objective.

Existing grounding work has demonstrated the benefits of grounding in improving the reasoning capabilities of LLMs, with two main directions emerging. The first direction augments the system with a separate external model [410, 221, 416]; these are used to generate explicit modality-aware outputs which are fed into an LLM. While these works provide insights into the value of non-text modalities, they are inherently limited by their external model and masks rather than addresses the lack of grounding in the underlying language model. This is an important distinction because more complex and abstracted concepts may be difficult to simulate, but may still require the foundational grounded components to correctly interpret. The second line of work used for grounding is the use of additional modalities during training to align internal representations [148, 365, 368]. These provide a direct signal to language models to improve their alignment with the environment. However these works rely on implicitly improving internal alignment; at inference, there is no clear representation of the environment to leverage. Further, these works focus on encoder-only models. Notably, modern LLM architecture has favored decoder architectures as the ability to generate open-ended outputs vastly increases the range of tasks they can accomplish.

RESEED combines the strengths of implicit alignment and explicit representations by training an LLM to predict latent state representations, which are then refed to the LLM to guide the language generation in a way that reflects with the true state of the environment. This approach provides the foundation for a scalable, grounded language model that operates in a manner consistent with modern LLMs. To evaluate RESEED, we require datasets that have paired text–trajectory data for training, but can test language models on text-only tasks. As these requirements cannot be found in existing benchmarks, we introduce three sequential reasoning datasets focused on cardinal direction navigation (ABCDs), block stacking (CUBES), and household object interactions (HOUSE). These tasks span increasingly complex state and action spaces. Compared to a text-only baseline, RESEED yields substantial gains in generalization and sample efficiency in sequential reasoning tasks.

Our contributions are: 1) RESEED, a novel grounding mechanism for decoder LLMs; 2) three new sequential reasoning benchmarks; 3) empirical validation of ReSEED, demonstrating improved sample efficiency and generalizability; and 4) ablations analyzing the components of ReSEED.

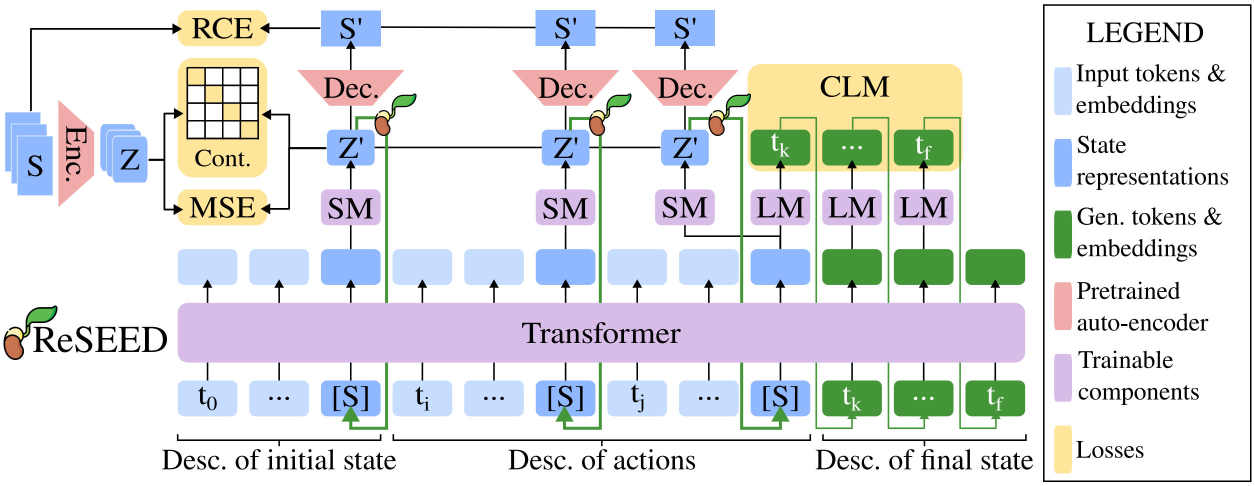


Figure 3.8: Architecture of ReSEED. ReSEED is comprised of a transformer with a language modeling head (LM) and a state modeling (SM) head (in purple). It leverages a pre-trained and frozen state auto-encoder (in red) during training. ReSEED requires two forward passes. The first pass (in blue) encodes the special *[S]* input tokens and uses the output of these tokens to generate state representations Z'. In the second pass (in green), the special tokens are replaced with linear projections of Z', which are used to generate the description of the final state. The alignment of Z' is trained using a Reconstructive Cross-Entropy (RCE), a Contrastive (Cont.), and a Mean Squared-Error (MSE) loss (in yellow). A Causal Language modeling (CLM) is used to train the generation.

3.3.2 Related Work

3.3.2.1 Grounding with External Models

A subset of existing systems enhance language-based reasoning by incorporating external modality-specific models. [394] leverages CLIP [295] to retrieve relevant images which are used to improve question answering. [367, 410] remove the need for an image database by using text-to-image diffusion models, while [421] directly leverages CLIP's text-model embeddings. While images offer rich *spatial* information, they cannot properly capture *temporal* information, which is key to sequential reasoning. To address this, [221] feeds outputs from a physics simulation engine into an LLM to improve physical reasoning. In all these approaches, language models are augmented with other modalities, rather than grounded to other modalities. We believe this distinction is critical, as we posit that grounded models can compositionally build on observed interactions, whereas augmented models face end-to-end training challenges and are constrained by the capacity of their external modules. PIGLeT [416] partially addresses these issues by using a trainable action prediction module to reason about household tasks. However, PIGLeT requires access to the ground-truth start state and only performs single-step reasoning. In contrast, ReSEED operates on text-alone and is designed for multi-step reasoning.

3.3.2.2 Grounding through Internal Alignment

A complementary line of work focuses on aligning an LLM's internal representations across text and auxiliary modalities. Like ReSEED, these methods use additional cross-modal modules during training, which are then discarded. We refer to these as implicit internal alignment methods.

Certain approaches in this space use additional modalities to produce more relevant text data. [58] adapts the BabyAI environment [66] to a text-based version, giving LLMs the ability to explore the environment in text. [406] generates goal-oriented and random exploration experiences in VirtualHome [290], and uses templates to create a home-navigation fine-tuning dataset. [203] create state annotations in TextWorld [76] and TRIP [353] to generate more coherent outputs. However, these methods remain limited by the abstraction and reporting bias inherent in text data.

Other approaches incorporate auxiliary losses conditioned on other modalities. [365] adds a visual token (voken) classification objective in pre-training. [148] introduces a cross-modal adaptation phase with joint MLM, voken classification, and image-text matching. Most similar to our approach, [368] train a teacher model using MLM and a contrastive cosine similarity task between video and text embeddings and then distill this knowledge into a student model. [169] combine the voken classification and distillation tasks to further improve results. However, these methods are all designed for encoder-only architectures, which are not well-suited for text generation. In contrast, ReSEED is developed for generative decoder-based models. More importantly, we identified that during implicit internal alignment, ReSEED was producing embeddings that were aligned with the state of the environment, and that these could be effectively re-used rather than being discarded.

3.3.3 Method

Our method, Refeeding State Embeddings aligned using Environmental Data (ReSEED), is depicted in Figure 3.8. It can be broken down into three stages: 1) pre-training a state auto-encoder (Section 3.3.3.2), 2) generating latent state representations using special tokens (Section 3.3.3.3) 3) re-feeding these tokens before generating the output (Section 3.3.3.4).

3.3.3.1 Prerequisites

ReSEED requires access to paired text-trajectory data. Specifically, for a given sequence of states ($s \in \{s_0, s_1, ..., s_f\}$), there should be a text description of the initial state ($d_0 \leftrightsquigarrow s_0$), a description of actions applied ($d_i \leftrightsquigarrow \Delta(s_{i-1}, s_i)$), and a description of the final state ($d_f \leftrightsquigarrow s_f$). In Section 3.3.3.5 we outline the datasets we use.

3.3.3.2 State Auto-Encoder

To create salient latent representations of the states, Z in our environment, we first train an auto-encoder (AE) using a reconstruction loss. Our AE is comprised of a 3-layer encoder multi-layer perceptron (MLP) and a 3-layer decoder MLP, both with dropout and trained using a cross-entropy reconstruction loss. The size of the latent representations is a hyperparameter h_{dim}, which we sweep $h_{dim} \in \{16, 64, 128, 256, 512\}$ for each dataset. We freeze the parameters of the AE when training the LLM and discard it after training is completed.

3.3.3.3 Generating Latent Representations

Our grounded language model adopts the convention of modern LLMs as a causal transformer. Given a description of the initial state and a sequence of actions, the model is trained to infer its own latent representation of the resulting states, denoted as Z', which should align with the true latent states Z. To enable this, we inject a special token *[S]* after each input description d_i. The corresponding output embedding is passed through a single-layer state modeling head, projecting it to h_{dim}. We additionally pass the produced latent state through the pre-trained decoder to produce a prediction of the full state, $S' = Dec(Z')$.

To guide alignment, we apply three complementary losses: a contrastive (Cont.) loss [265] between Z' and Z, a mean-squared error (MSE) loss between Z' and Z, and reconstruction cross-entropy (RCE) loss between S' and S.

$$\mathcal{L}_{Cont.} = \mathbb{E}_i \left[-\log \frac{\exp(\text{sim}(Z'_i, Z_i)/\tau)}{\sum_{j=1}^{N} \exp(\text{sim}(Z'_i, Z_j)/\tau)} \right]$$

$$\mathcal{L}_{MSE} = \mathbb{E}_i \left[\|Z'_i - Z_i\|_2^2 \right]$$

$$\mathcal{L}_{RCE} = \mathbb{E}_i \left[-\sum_m S_i^{(m)} \log \left(S_i'^{(m)} \right) \right]$$

where $N = \sum_{k=1}^{B} |S_k|$, B is the batch size and $|S_k|$ is the number of states in sequence k, i.e., we use in-batch and in-sequence negatives in our contrastive loss.[17] A comparison of the impact of each loss is shown in Table 3.16.

3.3.3.4 Refeeding Embeddings

Sections 3.3.3.2 and 3.3.3.3 produce an LLM that is implicitly aligned and capable of generating salient latent representations of states. Motivated by the idea that these latent representations carry useful information about the environment, we develop a *refeeding* mechanism, in which a second forward pass is performed with the special *[S]* tokens being replaced with linear projections of Z'. This enables the model to explicitly condition its generation on its own representation of the environment. On this second pass we apply the traditional causal language modeling loss on the final state description:

$$\mathcal{L}_{CLM} = -\sum_{t=k}^{T} \log P(x_t \mid x_{<t})$$

where k indexes of the first token of the final state description and T is the total number of tokens.

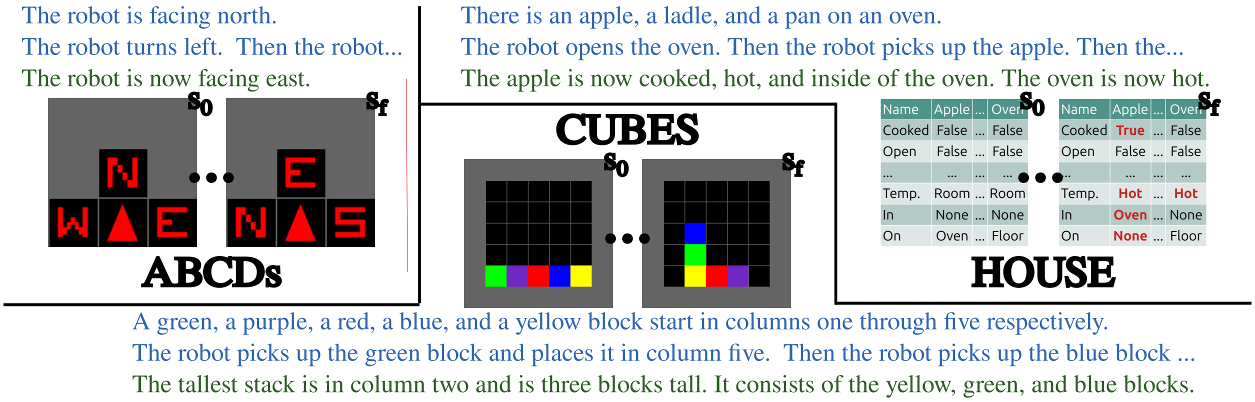
We note here three clear differences with the most related work of VidLanKD [368]. The first is the use of a causal language modeling which enables text generation. Second, unlike VidLanKD that uses a single

---

[17] i, j, k are overloaded and used as general indexing terms.

embedding to encode the entire sequence, we leverage separate embeddings for each timestep in the sequence. This provides two benefits: 1) it allows the LLM to align itself multiple times per sequence, providing a denser learning signal, and 2) it provides more useful negatives in the contrastive loss as the model has to identify the impact of the actions to be able to differentiate different states from the same sequence. Without these more difficult negatives, the model may be able to rely on more surface level features—e.g., the objects in the scene—to differentiate embeddings and lose the specificity required for successful grounding. Third, instead of relying solely on implicit internal alignment, the refeeding provides an explicit mechanism to make use of our aligned representations. We report the impact using multiple state representations and explicit refeeding in Tables 3.17 and 3.18, respectively.

3.3.3.5 Datasets



The robot is facing north.
The robot turns left. Then the robot...
The robot is now facing east.

There is an apple, a ladle, and a pan on an oven.
The robot opens the oven. Then the robot picks up the apple. Then the...
The apple is now cooked, hot, and inside of the oven. The oven is now hot.

Name	Apple	...	Oven
Cooked	False	...	False
Open	False	...	False
...	...	...	...
Temp.	Room	...	Room
In	None	...	None
On	Oven	...	Floor

Name	Apple	...	Oven
Cooked	**True**	...	False
Open	False	...	False
...	...	...	...
Temp.	**Hot**	...	**Hot**
In	**Oven**	...	None
On	**None**	...	Floor

A green, a purple, a red, a blue, and a yellow block start in columns one through five respectively.
The robot picks up the green block and places it in column five. Then the robot picks up the blue block ...
The tallest stack is in column two and is three blocks tall. It consists of the yellow, green, and blue blocks.

Figure 3.9: A sample from the ABCDs, CUBES, and HOUSE datasets. The blue text defines the initial state (s_o) and the actions performed (truncated for space). The orange text defines the final state (s_f). The model is also provided with access to intermediate states, which are collapsed into ellipses in the figure due to space.

ReSEED is designed to leverage the rich information found in environments during training, while relying solely on text during inference. Naturally, this requires datasets that provide paired natural language and trajectory data for training, along with language-only evaluation sets. While prior work in natural language task specification—such as [246, 417, 73]—offers partially aligned training data, their evaluation protocols remain grounded in agent-based execution, lacking the necessary text-only test conditions. To bridge this gap, we introduce three new datasets that span distinct domains: cardinal direction navigation, block stacking, and interaction with common household objects. We describe each in detail below, and an example question and trajectory of each is shown in Figure 3.9.

ABCDs: Asking 'Bout Cardinal Directions The first domain requires an model to understand navigation of cardinal directions. In ABCDs, an agent starts facing one of the four cardinal directions—$\{$North, East, South, West$\}$—then the agent performs a sequences of turns, and is then asked which direction it is facing at the completion of all turns. To create the text component of the dataset we use a template and create a mapping between a set of natural language action phrases and the equivalent base action. For example, the action phrase "turn 270 degrees clockwise" would map to the action turn left. We can then combine a description of the start state with a sequence of A action phrases, and a description of the end state. For the trajectories, we create a small grid environment using gym-minigrid [67] with compass-style markers in the walls to indicate the direction and use an egocentric grid representation to encode each state.

For training and validation, we use up to $A \leq 5$ action phrases. To evaluate length generalization, we construct five evaluation sets, each containing 2000 samples and using a fixed number of action phrases, with $A \in 6, 7, 8, 9, 10$. We report exact match accuracy on each of these held-out sets.

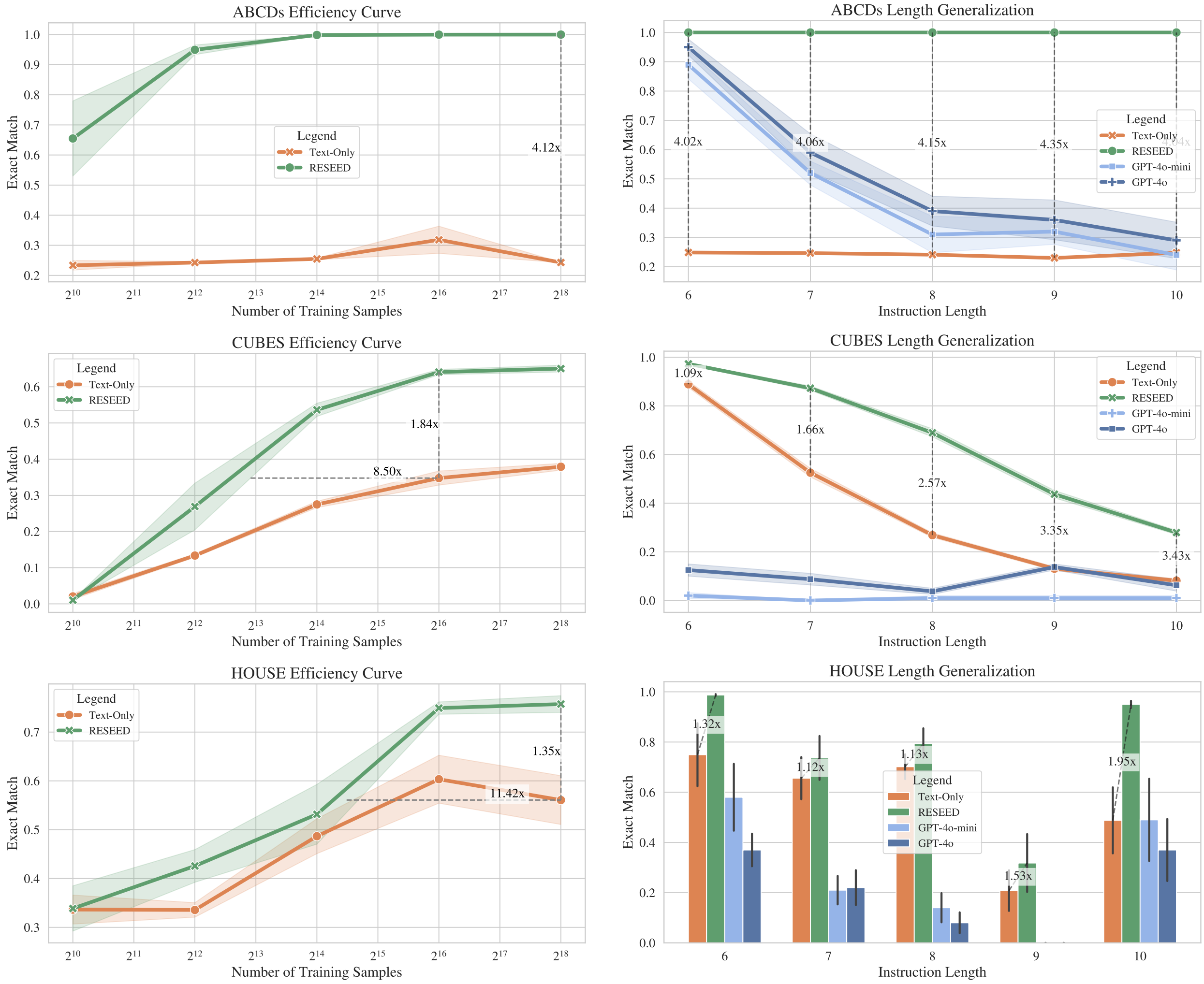
This dataset provides a test bed where the state and action spaces are small, with only four different observations and underlying actions. This leads to a domain where the syntax of the language is similar and difficult to distinguish, but the semantics contained within the trajectory are clear and easily distinguishable.

Comprehensive Understanding of Block-stacking and Effects of Sequences (CUBES) CUBES tests a models ability to identify the tallest stack of blocks after a sequence of stacking actions. An initial state is presented with five different-colored blocks in a random order. A series of A stacks are then performed. Similarly to ABCDs, we use templates for the language component and gym-minigrid to create paired state trajectories. Unlike ABCDs, we use a full view of the all the blocks.

We match the setup for ABCD, with $A \leq 5$ action phrases for training and validation, and five length generalization sets which use $6 \leq A \leq 10$. We report the exact match accuracy on these length generalization sets.

Compared to ABCDs, the state space and action spaces are significantly larger, however the language still only requires a small vocabulary. The syntax of the language is still difficult to distinguish, however the semantics contained within the trajectory are less distinguishable than in ABCDs.

HOUSE: Household Object Use in Sequential Execution HOUSE is inspired by the PigPen dataset used in the Piglet framework [416]. In this dataset, a series of tasks are carried out using 100 common household objects with varying affordances. HOUSE consists of 9 atomic actions (e.g., pick up object, toggle object on, ...), which we compose into 10 low-level tasks (e.g., put X in Y, heat X, ...) and 10 high-level tasks (e.g., 'brew tea', 'water plants', ...). The low-level tasks are comprised of 2-5 atomic actions, while the high-level tasks are themselves composed of 2-3 low level tasks with a total of 6-10 atomic actions. Each task uses up to four objects, and the state space is defined as the state of the four objects, including the object name and the current features of the objects. A full description of the dataset, including a comparison to PigPen, the set of atomic actions, low-level tasks, and high-level tasks is outlined in Appendix C.1.



(a) Sample efficiency evaluation

(b) Length generalization evaluation

Figure 3.10: Sample efficiency and length generalization results on the three benchmarks.

Mirroring ABCDs and CUBES, we train and validate using the low-level tasks, which include sequences of $2 \leq A \leq 5$, and evaluate the LLMs on the high-level tasks. There are two high-level tasks for each $A \in 6, 7, 8, 9, 10$, and we use 1000 samples per high-level task. We report the exact match accuracy on the high-level task sets.

HOUSE provides a step toward more general tasks that includes a wide range of objects and actions. Compared to ABCD and CUBES, the vocabulary, action space, and state space are all larger, which increases difficulty. However, the syntatic variation is also larger, making the impact of actions more apparent.

Lastly, whereas ABCDs and CUBES evaluate length generalization by repeatedly applying the same kind of actions, HOUSE's evaluations requires compositionally applying observed sequences, which is an additional challenge.

3.3.3.6 Experimental Setup

The baseline for our experiments is a text-only (TO) model that shares the same underlying transformer and language modeling head and only differs in its lack of a state modeling head and projection layer. With an h_{dim} of 256 (the largest used across our datasets), RESEED only uses 0.4% more parameters (83.5M vs 83.9M parameters). The *TO* model is trained using the standard causal language modeling loss. For both RESEED and *TO*, we initialize the transformer and language modeling head using HuggingFace's [402] pretrained gpt2-base [297]. The state modeling head is randomly initialized. We then finetune the model on the datasets until convergence is reached on a validation set that is 12.5% $(1/8^{th})$ the size of the training set. We define convergence as having no improvement for more than 5 epochs. To enable evaluation context lengths that are longer than those seen in the training dataset, we freeze the positions ids of the pre-trained gpt2. We train the models using an AdamW optimizer [226] with an exponential learning rate decay. We tuned the *TO* model using a grid search on the learning rate, batch size, decay rate, and warm up steps, and used the same parameters for RESEED. We run each experiment using five random seeds, and report the standard error across seeds. Additional details can be found in Appendix C.4. We will open source all code at the end of the anonymity period.

3.3.4 Research Questions

RQ1: Does the grounding provided by* RESEED *improve sample efficiency? As RESEED has access to additional rich and unabstracted information during training in the form of environment data, we hypothesize that RESEED will be more sample efficient than an LLM trained solely on text, *TO*. To test this, for each dataset, we generate six different training sets, with number of samples $\in \{1024, 4096, 16384, 65536, 262144\}$ respectively.

RQ2: Does the grounding provided by* RESEED *improve length generalization? RESEED's generation of latent state representations, Z', enables it to produce estimates of the true states at regular intervals before decoding the final state, which we hypothesize will allow it to maintain a more consistent interpretation of the environment even in longer time horizons. To test this, we compare the results of ReSEED and *TO* on the different length generalization evaluation sets we created when trained on the full 2^{18} samples.

RQ3: Which alignment signal—contrastive, reconstructive, or mean square error—best grounds language? A crucial step in our process is the alignment of the latent state representations produced by the LLM with latent state produced from our auto-encoder. To this end, we compare ReSEED (RS for short), which uses all three losses, to variations that use one of the three alignment losses ($\mathrm{RS}_{\texttt{Cont}}$, $\mathrm{RS}_{\texttt{MSE}}$, $\mathrm{RS}_{\texttt{RCE}}$), and a variation which uses no alignment loss ($\mathrm{RS}_{\texttt{None}}$). This last variation is equivalent to providing the refeeding mechanism to the *TO* baseline.

RQ4: Does providing alignment at each state improve grounding? One of the core differences with [368] is the alignment of our model at each state compared to a single alignment per text/state sequence pair. To understand the impact of this difference, we generate three variations of each dataset. The first uses the existing setup, where *[S]* tokens are added for each state in the ground-truth trajectory. The second uses a *[S]* token only for the first and last state in the true trajectory. The third, only uses a *[S]* token for the last state; this final variation most closely resembles the token setup in [368], albeit for a decoder transformer.

RQ5: How beneficial is explicit refeeding compared to implicit alignment? A second core difference with [368] is our method of explicitly refeeding the aligned representations before decoding. To ablate the benefits of explicit refeeding, we compare ReSEED with an implicitly aligned version that is trained using all the same losses, but only performs a single forward pass.

3.3.5 Results & Discussion

RQ1: Does the grounding provided by* ReSEED *improve sample efficiency? The graphs on the left side of Figure 3.10 show the average results across all evaluation splits of the text-only baseline (in orange) and of ReSEED (in green). While there is a small benefit when using a small amount of data, the benefit continues to grow larger after this point. Notably, once a minimum amount of data is reached, ReSEED is able to leverage the environment data to improve upon text-only training. This leads to ReSEED scaling better than than text-only training, which is an extremely promising result.

RQ2: Does the grounding provided by* ReSEED *improve length generalization? The graphs on the right side of Figure 3.10 show the results on each of the evaluation splits when using all training samples (2^{18}) for both the text-only baseline (in orange) and of ReSEED (in green). Once again, we see ReSEED outperforming the text-only baseline on every evaluation split. In addition to instruction length, the range of the underlying atomic actions and low-level tasks in the HOUSE dataset directly impact the complexity of the high-level tasks, adding an additional dimension to the evaluation splits. For this reason, we present the results in a bar chart rather than the line chart used in ABCDs and CUBES. The varying complexity of actions also explains the noisier trends seen in the HOUSE dataset results.

Model	ABCDs	CUBES	HOUSE
`Text-Only`	24.3 ±0.1	37.9 ±0.9	56.1 ±5.0
$\texttt{RS}_{\texttt{None}}$	12.1 ±4.0	39.4 ±1.5	52.7 ±6.1
$\texttt{RS}_{\texttt{Cont}}$	10.8 ±4.3	**66.5** ±1.1	68.2 ±2.4
$\texttt{RS}_{\texttt{MSE}}$	99.7 ±0.3	33.4 ±1.3	60.7 ±4.1
$\texttt{RS}_{\texttt{RCE}}$	81.0 ±19.0	59.7 ±1.8	68.2 ±3.1
$\texttt{RS}_{\texttt{All3}}$	**100.0** ±0.0	65.0 ±0.9	**75.7** ±1.7

Table 3.16: Comparison of alignment losses used in RESEED (RS). `All3` indicates a combination of all three alignment losses. Results are the avg. accuracy and std. error across 5 seeds.

RQ3: Which alignment signal—contrastive, reconstructive, or mean square error—best grounds language? From Table 3.16, we see that alignment is necessary; $\texttt{RS}_{None}$ performs similarly to the *TO* model. Interestingly, the alignment signal that is the most beneficial varies per dataset, but using all alignment signals, $\texttt{RS}_{All3}$, provides competitive results in all three datasets. As such, this is the setup we use for all other experiments.

Model	ABCDs	CUBES	HOUSE
$\texttt{TO}_{\texttt{Final}}$	24.6 ± 0.2	37.3 ± 0.9	69.9 ± 1.1
$\texttt{TO}_{\texttt{Init\&Final}}$	24.5 ± 0.1	36.6 ± 1.1	68.7 ± 1.1
$\texttt{TO}_{\texttt{Per Phrase}}$	24.3 ± 0.1	37.9 ± 0.9	56.1 ± 5.0
$\texttt{RS}_{\texttt{Final}}$	24.2 ± 0.0	61.9 ± 1.1	68.1 ± 2.9
$\texttt{RS}_{\texttt{Init\&Final}}$	33.0 ± 8.6	62.7 ± 1.9	71.9 ± 3.4
$\texttt{RS}_{\texttt{Per Phrase}}$	**100.0** ± 0.0	**65.0** ± 0.9	**75.7** ± 1.7

Table 3.17: Comparison of RESEED (RS) and a text-only (TO) baseline with varying [S] token frequencies. Results are the average accuracy and standard error across 5 seeds.

RQ4: Does providing alignment at each state improve grounding? From Table 3.17, for the text-only baseline, including additional *[S]* tokens either has minimal impact, or, in the case of HOUSE, is deteriorates performance. In this latter case, we hypothesize the additional token(s) can be used by the model to further overfit to the training data. In contrast, for RESEED, we see a clear trend of improvement when including additional state representations, with $\texttt{RS}_{\texttt{Per Phrase}}$ providing the best result and lowest standard error in each dataset.

Model	ABCDs	CUBES	HOUSE
`Text-Only`	24.3 ± 0.1	37.9 ± 0.9	56.1 ± 5.0
$\texttt{RS}_{\texttt{Implicit}}$	24.5 ± 0.1	34.3 ± 3.1	59.1 ± 4.3
$\texttt{RS}_{\texttt{Explicit}}$	**100.0** ± 0.0	**65.0** ± 0.9	**75.7** ± 1.7

Table 3.18: Comparison of RESEED with and without explicit refeeding. Results are the average accuracy and standard error across 5 seeds.

RQ5: How beneficial is explicit refeeding compared to implicit alignment? Table 3.18 demonstrates that explicitly refeeding the learned representations is core to the performance of RESEED. Unlike prior work, implicit alignment provides little to no benefit in our experiments. As the system was tuned for explicit refeeding, it is possible that implicit alignment could be improved if different subsets of losses or hyper parameters are used, or if additional methods, such as [368]'s teacher-student distillation, are integrated. However, given the more direct signal it provides and the results in Table 3.18, we believe explicit refeeding is a stronger mechanism to ground language. We note that refeeding does comes at the cost of a second forward pass, increasing compute and training time. However, this is a relatively small cost for improved generalizability of the model.

3.3.5.1 Comparison to State of the Art

The primary motivation of this paper was the inherent limitations of text-only training. To this end, we evaluate several State-of-the-Art LLMs from the Qwen2.5 [408] and GPT4o [267] family on our benchmarks. In each instance, we provide a prompt describing the task and provide 10 in-context examples. The full prompt can be found in Appendix C.2. We report the results in Table 3.19. The results are in line with other work [91, 379], which demonstrate that current text-only LLMs struggle on tasks involving multi-step reasoning. Notably RESEED outperforms every model on every dataset, while being orders of magnitudes smaller.

Model	Size	ABCDs	CUBES	HOUSE
`RESEED`	84M	**100.0** ± 0.0	**65.0** ± 0.9	**75.7** ± 1.7
`Qwen2.5`	0.5B	31.4 ± 1.0	0.2 ± 0.2	1.8 ± 0.6
`Qwen2.5`	3B	42.8 ± 1.6	0.6 ± 0.2	4.0 ± 0.5
`Qwen2.5`	7B	40.0 ± 2.2	0.4 ± 0.4	26.4 ± 0.4
`GPT4o`	mini	45.6 ± 2.1	1.0 ± 0.5	28.4 ± 1.8
`GPT4o`		51.6 ± 2.5	9.0 ± 0.7	20.8 ± 1.4

Table 3.19: Comparison of RESEED (RS) to modern LLMs. Modern LLMs are provided 10 in-context examples and are evaluated on a subset of 100 examples divided evenly across evaluation splits. Results are the average accuracy and standard error across 5 seeds.

3.3.6 Conclusion

In this work, we introduce RESEED, a grounding mechanism that enhances sequential reasoning in LLMs by generating and refeeding latent state embeddings derived from environmental feedback. Through evaluation on three newly developed benchmarks—ABCDs, CUBES, and HOUSE—we show that

ReSEEDsignificantly improves generalization to longer instruction sequences and outperforms text-only baselines in both accuracy and scalability. These findings underscore the value of structured, grounded signals in supporting robust generalization beyond the capabilities of text-only training. However, progress in this area is currently constrained by the lack of high-quality, paired text-trajectory datasets. To advance the development of models that reason effectively over actions and states in real-world contexts, we highlight the urgent need for diverse, richly annotated datasets that reflect the structured and multimodal nature of human experience.

3.3.7 Limitations

ReSEED faces two primary limitations.

First, it introduces additional computational overhead due to the need for two forward passes. This cost is most significant during training, as the longer gradient path requires more memory and the addtional forward pass increase the time taken to complete one epoch. At inference time, no gradients are used used and iterative generation is already standard. To mitigate memory requirements, we explored a two-stage optimization procedure: one forward and backward pass to align latent states, followed by a second separate forward and backward pass to train generation. As shown in Appendix C.3, this approach still outperforms the baseline and only slightly underperforms the one-stage procedure, making a viable alternative is memory constraints exist.

Second, ReSEED requires access to paired text-trajectory data for training. While this limits applicability in domains lacking such resources, our results demonstrate the substantial value of this supervision signal. We hope this work encourages the development of more diverse and scalable text-trajectory datasets, and we view this as a necessary step for progress in grounded language understanding.

Finally, we note an additional limitation of our evaluation benchmarks, which while diverse in structure, are still limited in scope. All three are deterministic, template-based, and operate in relatively constrained state and action spaces. In contrast, real-world environments often involve ambiguity, stochasticity, and varied linguistic expression. Moreover, even our largest benchmark contains only 2^{18} examples—small relative to modern pretraining corpora. Extending ReSEED to broader, more complex, and non-deterministic domains is an important direction for future work. Doing so, however, will require scaling up dataset creation efforts accordingly.

3.4 Eyes on the Game: Deciphering Implicit Human Signals to Infer Human Proficiency, Trust, and Intent

The work described in this section has been published in ROMAN 2024 [159].

3.4.1 Introduction

With the continued advancement of artificial intelligence (AI) and robotics, it has become increasingly important to develop autonomous agents that can effectively collaborate with humans. One promising research direction focuses on endowing agents with a theory of mind [293], involving the development of mental models of teammates to improve adaptability [362]. *"Explicit"* communication—which is direct, unambiguous, and oftentimes verbal—can help form such mental models [313]. However, in many real-world teaming scenarios, only *"implicit"* communication—which is indirect, suggestive, and often non-verbal—may be possible. This could be due to factors such as the need for rapid action execution or high levels of ambient noise. In these scenarios, autonomous agents must rely on implicit signals to understand their teammates. Two implicit signals have been identified in the literature as promising options for these scenarios: 1) a teammate's behavior in the environment [19], which informs about intent and can anticipate future behavior and 2) a persons's visual attention, which provides fine-grained, immediate signals about their focus [2]. While these signals have been leveraged independently to model and predict human behavior, few works have sought to combine them. In this work, we hypothesize that by integrating these streams a more nuanced and complete model of a teammate can be learned. It is worth noting that acquiring high-fidelity data of complex cooperative tasks in sufficient quantities for deep learning models and comprehensive analysis is still a challenge in the field. This paper not only provides such a dataset collected through a large user study, but also provides a state-of-the-art (SotA) framework in the form of a causal transformer [298] and analysis comparing different implicit signals to predict a human's: a) proficiency at a task; b) trust in an autonomous teammate; and c) future intents. Prior work in human-robot interaction (HRI) and human-computer interaction

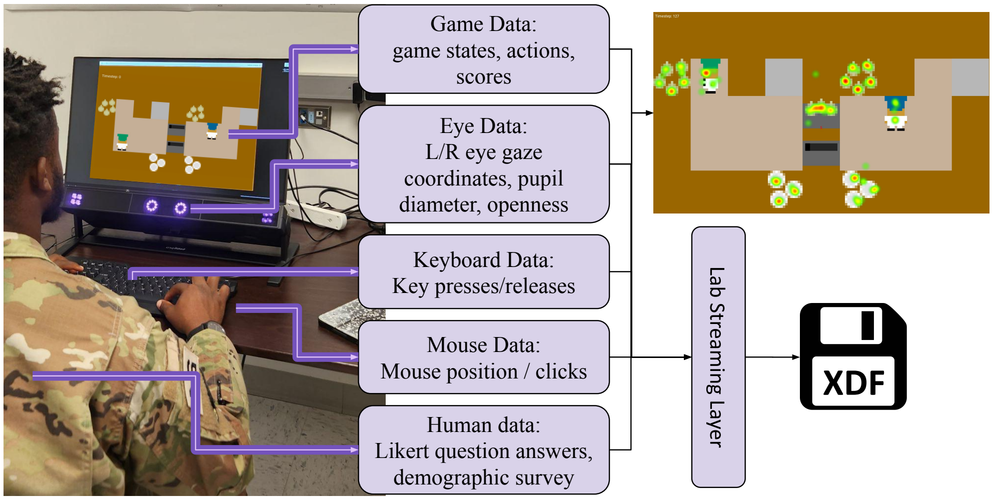


Figure 3.11: In this work, collect a large dataset of paired eye gaze and gameplay data in the collaborative game ”Overcooked.” Using this data, we train a causal transformer demonstrating state-of-the-art performance in its ability to predict a collaborator’s task proficiency, trust in an autonomous teammate, and future intent.

(HCI) have demonstrated the predictive power of implicit signals like eye gaze [422, 230, 306] and behavioral data [150, 154, 206]. Despite these advances, existing work still has limitations. First, most of these works infer only a single data point about their human teammates rather than build a comprehensive model of their behavior. Second, they are often applied to non-representative, turn-based environments where a single action can span several seconds and the autonomous agent is limited compared to the human teammate. Third, when eye gaze data is employed, it is often hand-crafted into a small set of features before being fed to predictive models, degrading a rich source of information for cluing the agent into the users' mental state. Finally, most of these papers do not publicly release their datasets, hindering replication, comparison, and further research.

In contrast, this work not only leverages eye-tracking and behavioral data *in parallel* to accurately predict multiple latent human factors, but also performs a comprehensive analysis of these inputs to determine the advantages of each data type. We collect this data in the fast-paced collaborative "Overcooked" environment (cf. see Figure 3.11 and [57]), which serves as an ideal testbed for human-AI teaming due to its capacity to efficiently gather large amounts of behavioral data in the form of intricate and coordinated gameplay at different levels of abstraction. We then leverage state-of-the-art deep learning models to predict multiple mental and behavioral factors including the human's intent (in the form of future attempted subtasks), their trust in the autonomous teammate, and their proficiency at the game. Additionally, we compare several methods to aggregate and represent eye gaze data, finding that gaze data provides salient information faster than gameplay data, but that gameplay provides a stronger signal as the task progresses. Combining the two consistently matches or outperforms the individual signals. Our results also show that gaze aggregation across the temporal dimension only minimally impacts results in our tasks, while the spatial aggregation method used in [387] substantially worsens performance.

In conclusion, we present the following contributions: 1) a time-series model that can be conditioned on both human eye gaze and gameplay data for accurate predictions of behavioral intents, skill level and trust in the agent; 2) a thorough analysis comparing the predictive power of gameplay data, eye gaze data, and the combination of the two, providing practical insights the contexts in which each type of data is most effective; and, 3) a publicly released dataset of gameplay data paired with eye gaze data in a fast-paced collaborative environment. We believe these insights derived will enable AI agents to better model human teammates, allowing faster and more specific adaptation to improve the team fluency and capability. By equipping agents with the ability to process implicit signals, we introduce new modes of understanding and expand the boundaries of human-agent interaction.

3.4.2 Related Work

Implicit human signals, such as EEG signals [170], heart rate [375], recent actions [150], body language cues [133], and eye gaze [2], have been studied as a means to improve the human-machine interaction [327].

* Human behavior as a predictor Human behavior often contains informative action cues that hint at future intent. For instance, reaching for a door handle suggests the intention to exit a room. Notably, past sequences of behavior have been used to improve human-robot collaboration on assembly tasks [391, 247], anticipate a human's action in a herding game [19], enhance human performance in teleoperation tasks [5], and predict a decision making in search and rescue [425].

Other work has investigated how to predict an action based on an observed initial portion of it. Wearable devices have been employed to collect arm movement data and improve prediction in handover tasks [393]. Progress has also been made on predicting actions from RGB images and optical flow [206] or RGB images alone [176], as well as on breaking down human movement into granular "movemes" to improve behavioral predictions [195].

* Eye gaze as a predictor Eye gaze stands out as a salient signal, providing rich insights into a person's attention, information processing, and social interactions, enhancing human teaming [436]. It has been used to anticipate intent in a robotic manipulation tasks [3], as a substitute for wake-words for smart-speakers [242], predict train routes in a turn-taking train board game [343], and detect errors in robot behavior [18]. Interestingly, the predictive power of gaze has been also demonstrated on the other end of the human-robot dyad: a robot equipped with a human-like binocular system and corresponding gaze controller [316] improves the human's ability to predict the robot's intent [44].

Our work shares similarities to [55] and [387]. Both works use implicit signals, including eye gaze, to predict information about humans in a fast-paced collaborative environment. [55] explores the use of gaze features, game data, survey data, and demographics to predict users' preferences between early game assistance and late game assistance. However, unlike our study, this work does not compare the use of gameplay data on its own to eye gaze data alone. [387] uses eye gaze data to predict periods of human confusion in the same environment we employ, however they do not consider the use of gameplay in any form. Notably, both of these works aggregate gaze data over both time and space. Our research differs by 1) thoroughly comparing gaze data to gameplay data, 2) examining the effects of aggregating gaze data in multiple different ways, and 3) exploring the predictive power of implicit communication across the three differentdimensions of trust, proficiency, and intent.

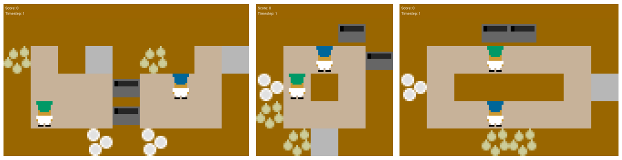
3.4.3 Method

3.4.3.1 Data Collection

Environment

Due to its highly flexibly nature and its ability to capture a wide-range of human-agent teaming behaviors, we focus our work on the collaborative cooking game "Overcooked" [57]. "Overcooked" requires a team composed of a human and an AI-controlled chefs to cook and serve as many soups as possible within a set time limit. To achieve this, players must execute a series of tasks ranging from collecting onions to placing them in a pot and serving the finished soups. Successful service rewards the team with 20 points. At each timestep, each player can choose one of the following base actions: *up*, *down*, *left*, *right*, *interact* with an object (to pick up, place, or serve items), or *stay* in place. "Overcooked" requires players to coordinate both on high-level strategic decision and on their underlying movements. At the strategic level, players should aim to minimize redundancy and inefficiency—for instance, avoiding the situation where both players retrieve a dish when only one soup is being prepared. On the movement level, careful navigation is essential to prevent collisions between players. This combination of strategic planning and movement precision makes "Overcooked" an especially suitable platform for studying human-agent collaboration.

Figure 3.12 shows the three specific game layouts we use to gather data: 1) Coordination Ring, which requires agents to focus heavily on their movement to avoid collisions 2) Asymmetric Advantages where agents are fully separated and so cannot collide, but must instead focus on aligning their high-level strategy, and 3) Counter Circuit that requires both movement and strategic alignment. Following previous work [13],



(a) Asymmetric Advantages (b) Coord. Ring (c) Counter Circuit

Figure 3.12: The three “Overcooked” layouts used. From [57].

we ran the experiments for 400 in-game timesteps at 5 FPS, which equates to 80 seconds of gameplay.

3.4.3.2 AI Agents

To capture a thorough and wide range of human behaviors, we collected data using three different agents of varying ability. The first agent is a random agent, which randomly selects one of the six base actions. This represents a very low level of play and is intended to create situations where the human may be confused about its teammate leading to low trust. The second agent is a self-play (SP) agent that is trained using reinforcement learning (RL)—specifically proximal policy optimization (PPO) [328]—and, as the name suggest, is trained being teamed with itself. This agent can be quite good at the game if the human adapts to its play-style, but its training regime causes it to be very rigid in its behaviors. This agent is aimed to create trials where the human can have moderate trust in their teammate, but must still pay attention to the agent’s behavior to avoid frequent collisions and a lower final score. Lastly, we use a SotA HAHA agent [13] that has been shown to be a significantly more performant, trusted, and understandable teammate. This agent was included to elicit situations where the humans have a high-level of trust in their teammate.

3.4.3.3 Trial Design

The primary objective in the dataset creation was to collect a wide range of human behaviors while performing a collaborative task from which we could analyze and compare the predictive ability of gameplay data and human eye gaze data. To this end, we ran an IRB-approved user study where we recorded participants playing the collaborative cooking game Overcooked. After completing consent forms, participants were

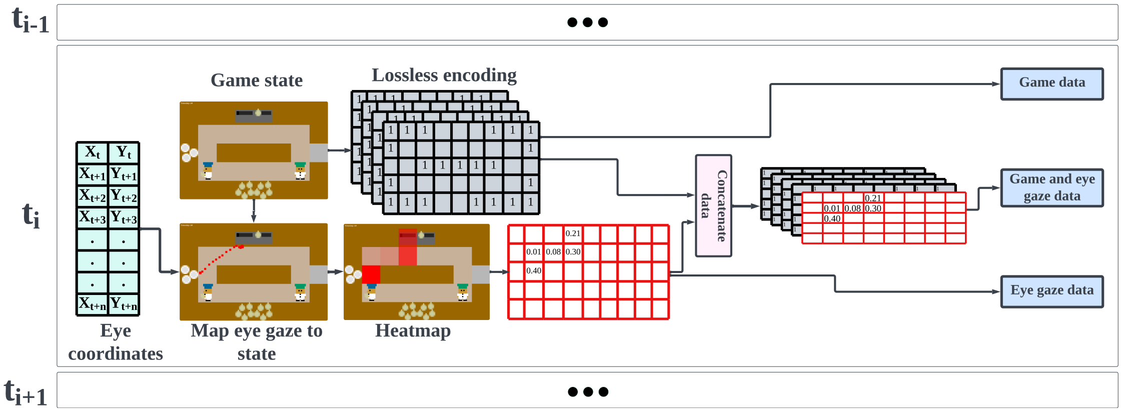


Figure 3.13: An overview of the processing method to create representations of eye gaze data, gameplay data, and enable a combination of the two for a single timestep. The representations are designed to be easily fed into modern neural networks.

required to fill out a demographic survey, read instructions about the game, and complete a short tutorial that required them to serve a completed soup before moving on. Each participant then played 18 rounds of Overcooked, with each round being played with one of three different agents on one of three different layouts. This led to each participant playing the nine different layout-agent combinations twice during the full duration of the trial. In the next section, we provide a more in depth description of the Overcooked environment and the specific layouts used, the set of different agents used, the number and recruitment methods for participants, the data collected during each trial, and how the data was processed.

3.4.3.4 Participants

In total, 83 participants were recruited across both the United States Air Force Academy (USAFA) and the University of Colorado (CU) Boulder using newsletter announcements and an online recruiting software. Nine participants were removed due to either technical difficulties with the system or poor eye tracking data quality ($> 40\%$ of eye tracking data was missing on at least one trial), leaving 74 participants in the dataset, for a total of 1332 total rounds of play or 29.6 hours of recorded play time. The age of participants ranged from 18 to 52 with an average age of 21.43. 33 participants identified as male, 39 as female, 1 non-binary, and 1 preferred not to disclose. When asked about their previous experience with Overcooked on a scale of 1 to 7 (1 being no experience), participants reported an average of 1.45, indicating that the majority of our participants had no or limited experience with the game. As our primary objective is to test predictive ability with unseen humans, we randomly select 59 participants for our training set, 5 participants for our validation set, and 10 participants for our test set. All participants were required to have normal vision ($20/40$ or better) without contact lenses to ensure that the eye capture system would be effective. Prior to participating, volunteers signed an informed consent document approved by the IRB at the Army Research Laboratory (ARL 23-079) in accordance with the Declaration of Helsinki.

3.4.3.5 User Study

Participant were required to complete an online demographic survey prior to their in person session. Upon arrival, participants signed a consent form, and were then positioned around 70cm away from a display and attached eye-tracking device (Tobii ProSpectrum), at which point a five-point calibration was executed using the Tobii Eye Tracker Manager (2.6.0) Figure 3.11. Subsequent to calibration, the accuracy of eye tracking was confirmed via real-time gaze tracking, with mandatory recalibration for any validation point discrepancies exceeding $1.5°$.

Following this, a Lab Streaming Layer (LSL) stream was started that broadcasted the eye gaze data (including, but not limited to, the right and left eye x and y coordinates relative to the display, as well as pupil dilation collected at $300Hz$), the game data (including, but not limited to, the game states, team actions, the reward, and instance of collisions), and the keyboard data and mouse data. All data was recorded in xdf files. See Figure 3.11 for depiction of the setup. Between each round, we additionally collected the participants answers on five statements adapted from [144] using a 7-point Likert scale [211] ranging from strong disagreement to strong agreement. These statements pertained to team fluency, perceived role significance, trust in the agent, understanding of agent actions, and the agent's cooperativeness.

3.4.3.6 Data Processing

To enable information to be readily fed into neural networks, we first clean and process collected data. We use [57]'s lossless state encoding function to encode the game states into a grid representation of shape height x width x 27, where each of the 27 channels contains information about different in-game objects or players. For the eye gaze data, we first average the x and y pixel coordinates of the two eyes. If the gaze data for a single eye is null for a given sample, we use the data from the only valid single eye data. If the gaze data is null for both eyes for a given sample, we exclude that sample. We then map the pixel coordinates to the corresponding tile in the game's grid environment. During this process, we filter out all eye gaze samples where the participant is not looking within the boundaries of the game environment. Since the eye gaze data is sampled at roughly 300Hz compared to the 5Hz (or FPS) of the gameplay data, we have approximately 60 eye gaze data points per gameplay timestep. To enable the combination of gameplay data and eye gaze data, we create eye gaze heatmaps of the same shape as the underlying game grid, and populate the grid with the ratio of gaze samples that fall within the boundaries of each tile. A visual representation of our method is shown in Figure 3.13. To compare our method to the method used in [387], we additionally map each eye gaze sample to a game grid tile and classify the sample as the human looking at their own agent, looking at their teammate, or looking at the environment. For each game timestep, we calculate the ratio of samples in each of these three bins.

After processing, we can readily use five input representations for a given number of timesteps. 1) a lossless game state encoding per game timestep (Game Data), 2) an eye gaze heatmap per game timestep (Eye Gaze Data), 3) a combined state encoding and the eye gaze heatmap per timestep (Game Data + Eye Gaze Data), 4) the average heatmap across all twenty timesteps (Collapsed Eye Gaze), and 5) the average ratio of eye gaze samples that map to the human's agent, teammate, environment across all timesteps (Gaze Object).

We use three different labels from our dataset. The first are the humans levels of agreement on the likert question: ”I trusted the agent to do the right thing:”. This ranges from 0 (strongly disagree) to 6 (strongly agree) with 3 being neutral. Second, for each agent-layout pair, we bin all scores in tertiles and label rounds by the tertiles they scored in. Scores in the bottom tertile would be in bin 0 (beginners), the middle tertile in bin 1 (intermediates), and the top tertile in bin 2 (experts). We use these score tertiles as a proxy for human proficiency. Third, we calculate when the human player completes one of the eleven different subtasks by identifying each time they perform an *interact* action and inspecting the change in state. We then back label each timestep since the previous subtask completion with the completed subtask. We use these subtask labels to predict a human's future intents. We note here that for trust and proficiency, there is a single label for a full 400 timestep round. For intent, there are many subtask labels in a single round, and the duration of a subtask label is highly variable and dependent on which task is being performed, the current layout, and the proficiency of the human.

As this data is collected from a human study and not hand curated, class distributions are not perfectly balanced. Reporting accuracy in this case can over state the performance. Due to this, we use an F1 score as our primary metric, as F1 scores incorporate both precision and recall in its final output. We additionally include a baseline model that always predicts the majority class in all our results.

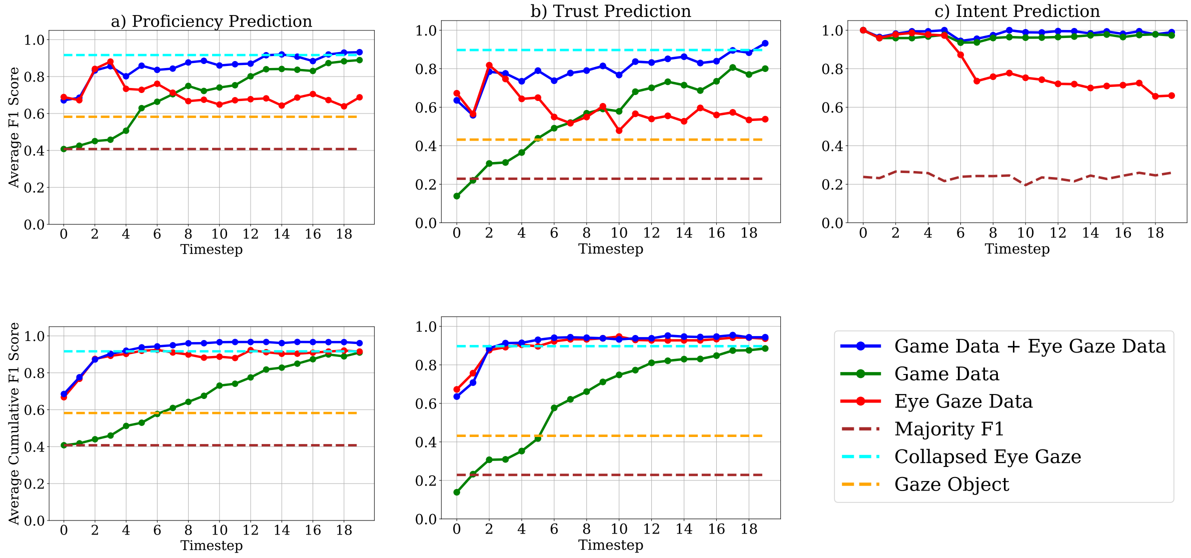


Figure 3.14: F1 scores over time for different implicit human signals predicting human proficiency, trust, and future intents starting at timestep **0** of each trial. The top row of graphs shows the per-timestep prediction outputted by our transformer model that can handle time-series data. The bottom row shows the cumulative prediction of all past timesteps. Dotted lines represent methods that aggregate over time and use the full 20 second window for their prediction.

3.4.4 Models

We train two types of models to predict our labels from the input data. For the input data types that retain time-series information—game data, eye gaze data, game data + eye gaze data—we first flatten the timestep representations, as in Figure 3.13. We then apply a linear layer to encode them into a token embedding size and pass the first 20 timesteps through a transformer model [381]. To capture the temporal dependencies in the data, we employ a causal transformer architecture [298]. Specifically, we generate a causal attention mask that ensures each output token can only attend to the previous tokens in the sequence. This masking mechanism is crucial for preventing information leakage from future timesteps and enabling the model to learn meaningful temporal patterns. To prevent overfitting, techniques like dropout and layer normalization are applied in positional encodings and transformer layers. Each output token is fed into a linear layer to get the appropriate number of logits for the task at hand. We use a cross-entropy loss between the logits and ground truth labels at every timesteps and the RAdam optimizer [220] to train the model. We use the same architecture parameters as the base model in [381]. We perform a grid search on learning rate: $lr \in \{1e-5, 3e-5, 1e-4\}$, batch size: $bs \in \{32, 64, 128\}$, warmup steps: $ws \in \{500, 1000, 2000\}$ and found $lr = 3e-5$, $bs = 128$, and $ws = 2000$ provided the best results.

For the two representations that aggregate over timesteps—collapsed eye gaze and gaze object—we average their representations over all 20 timesteps and then feed the aggregated input into a three layer multi-layered perceptron with 128 hidden units. We use the same loss function and optimizer. We perform the same grid search excluding warmup steps which are transformer specific and found $lr = 1e-4$ and $bs = 128$ provided the best results.

3.4.4.1 Data Release

An anonymized version of the collected data and the code used to process it can be found online[18] . The dataset contains XDF files that include all eye gaze data at 300Hz and all gameplay data at 5Hz. Additionally, they include keyboard and mouse data that were not utilized in our analysis. In addition to the XDF files, the datset contains the results of the likert scale questions, which can be mapped to the XDF files using anonymized user and trial ids.

3.4.5 Experimental Design

With the collected data, we set out to answer the following three research questions. **RQ1: How does the predictive power of eye gaze data compare to the predictive power of gameplay data and to the combination of both?** Core to our contributions is a thorough analysis of the predictive power of gaze data compared to gameplay data. To this end, we train a causal model per agent-layout combination on the first 20 timesteps of each round for each of our three prediction labels: trust, proficiency, and next subtask to be completed.

RQ2: How does aggregating eye gaze data along spatial and temporal dimensions effect its

---

[18] `https://hiro-group.ronc.one/overcooked-eye-gazedataset` hosts the dataset. `https://github.com/HIRO-group/HAHA/tree/EyeGaze` hosts the code used to process this data.

predictive power? Recent work has often aggregated eye gaze data across different dimensions to simplify the input space [387, 55]. This immediately poses the question of if and by how much these simplifying aggregation techniques are impacting the predictive power of eye gaze data. To test this, we compare the predictive power when using the full time series eye gaze data to two lossy methods. In the first method, we average the heatmap across timesteps, which collapses the temporal dimension of the data and that we name *collapsed eye gaze*. In the second, inspired by the approach used in [387], we collapses the spatial dimension and only looks at the ratios of eye focus on the user themselves, the teammate, and the environment. We name this method *gaze objects*.

RQ3: Does the predictive power of eye gaze and gameplay data differ between the start of a new task and during continuous execution? Lastly, we hypothesize that a human's work flow may change between the start of a new task and when they have been performing the same task for a while. If true, we expect to see a difference in game play and gaze data patterns. To examine this, we compare the predictive power of eye gaze and gameplay data on when focusing on the first 20 timesteps of gameplay compared to focusing on timesteps 200 to 220.

3.4.6 Results

RQ1: Comparing eye gaze data to gameplay data. Section 3.4.3.6 depicts the predictive power of eye gaze data, gameplay data, and their combination across multiple human mental and behavioral factors. We first focus on the intent, or "next subtask" prediction, shown in Section 3.4.3.6 c). As this particular analysis only considers the inital 20 time steps of the game, almost all participants will retrieve an onion as their first subtask, leading to a very high f1 score early on. However, whereas the models that use game data and the combination of game and gaze data maintain a high predictive ability, using ungrounded gaze data on its own leads to a drop in performance at later stages. This is likely because the gaze data only provides information on *where* the human is looking, but without the game data, there is no information on *what* the human is looking at. While the model can memorize the location of fixed objects in the environment to provide a better than random prediction, there is no way to know where either agent is situated or to model the dynamic changes to the environment. In a situation where the human looks at the location where a pot is, for example, from ungrounded eye data alone it would be difficult to ascertain whether the human is going to drop an onion into the pot, retrieve a completed soup from the pot, or perhaps check if their teammate is performing either of these actions.

We next focus on the proficiency and trust predictions, shown in Section 3.4.3.6 a) and b). For these, since the labels are identical for an entire round, we provide two versions of the graph. The top row of graphs show the individual predictions at each separate timestep, whereas the bottom row of graphs averages all the probabilities up to and including the current timestep. These results show a clear trend where the predictive power of gameplay data starts near majority class prediction, and consistently trends upwards. This is expected because the game's initial state is the same in all trials, and the model can refine its prediction as trajectories deviate toward or away from optimal paths. Notably, game data achieves a high performance within 20 timesteps.

Eye gaze data alone provides a strong predictive signal very early on, spiking around timestep 3 and 4 in both the trust and proficiency predictions. A qualitative analysis of the eye gaze and behavior at the start of the game showed a trend where the participants would first look at their teammate, then switch their focus to their agent before performing their first productive action. For more experienced players, this shift would occur around this threshold of timestep $3 - 4$, whereas for less experienced players, it would occur later, usually between timesteps 7 and 15, aligning with the results we see here. Similarly to the subtask prediction, as the players deviate away from the start state, the gaze data lacks necessary game information and the models start to lose some of its predictive ability. However, as seen in the bottom row, this can be significantly mitigated by using the cumulative probabilities of all predictions.

Lastly, using a combination of eye gaze data and gameplay data provides the best of both modalities, requiring little data to get a good performance, and continuing to improve with more data. Unlike the ungrounded gaze only models, the gaze here can be attributed to objects in the environment, and we see no drop in performance. In all cases, using both modalities provides the best or tied for best performance.

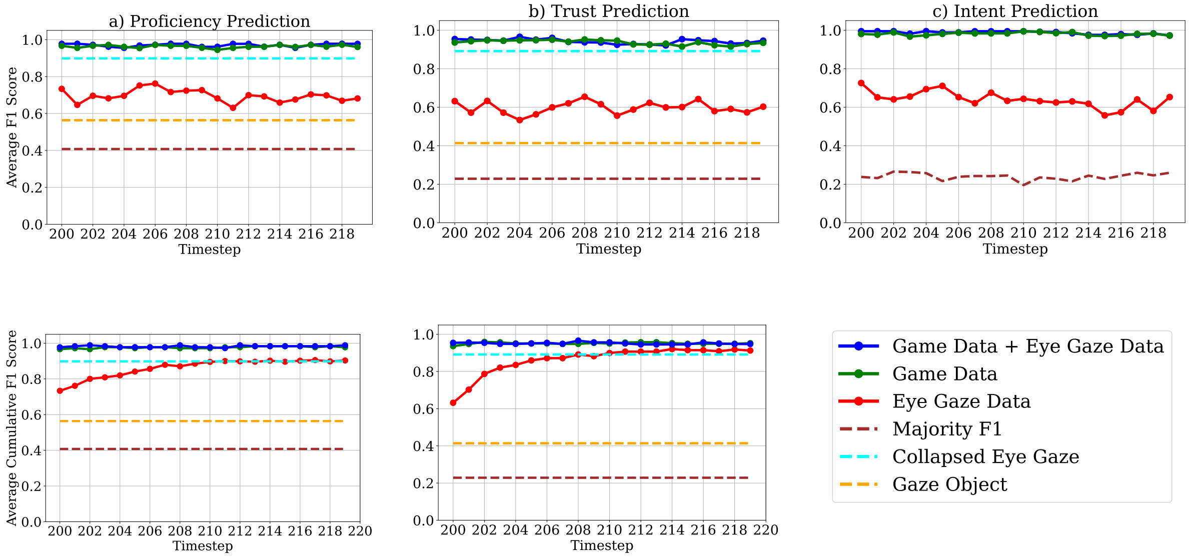


Figure 3.15: F1 scores starting at timestep **200**. Refer to Section 3.4.3.6 for a full description of the figure

RQ2: The effect of gaze data representation. We next investigate different methods to represent eye gaze data. Specifically, we compare the full time series representation utilized in the previous section to the collapsed gaze and gaze object representations. We note the latter two approaches use all timesteps in question for their prediction, and therefore are only comparable to the final timestep prediction. In the cumulative F1 approach (bottom graphs), we see that the time series approach matches or outperforms the other approaches. However, the collapsed eye gaze approach performs nearly as well using a simpler model. In contrast, the gaze object approach of [387], which collapses the spatial dimension and only uses the frequency at which humans look at different objects, drastically reduces performance. These results indicate that the spatial dimension of gaze data is more useful for predicting proficiency and trust compared to the temporal dimension.

RQ3: Task time We now examine how predictive power changes as humans move from starting a new tasks to a phase of continuous execution of the task. Figure 3.15 show the predictions curves when we start predicting at timestep 200. Compared to the previous results, there are two trends. First, gameplay data is now strongly predictive from the first observed timestep, indicating that the state space itself contains a significant amount of information about the quality of play up to that point. Second, we see that the per-timestep prediction of the ungrounded gaze model no longer has the spike in prediction accuracy at the beginning, but rather consistently predicts trust and proficiency at a similar level to its predictions around timesteps 20. Notably, even without this beneficial early bump, the cumulative prediction accuracy (bottom graphs) increases over time and achieves a substantially higher F1 score than any single timestep, indicating that even with no grounding, repeated measures of eye gaze data contain a rich signals about human behavior.

3.4.7 Discussion and Conclusion

In this work, we present a large-scale dataset of human gameplay and eye gaze collected during collaboration with a variety of agents in the Overcooked environment. Using this dataset, we investigate how implicit behavioral signals can be used to infer key human attributes relevant for effective human-agent interaction. Our analysis yields several key findings: 1)both eye gaze and gameplay data serve as strong predictors of human proficiency, trust, and intent; 2)eye gaze offers early predictive power, particularly at the outset of a task when limited behavioral data is available, while gameplay becomes more informative as the task unfolds; 3)combining gaze and gameplay yields the strongest overall inference performance; and 4) care must be taken when aggregating eye gaze data, with temporal aggregation proving more robust than spatial aggregation in our setting.

These results demonstrate that implicit signals—especially when integrated—are rich sources of information for modeling human behavior, supporting rapid adaptation and effective teaming. This mirrors how humans intuitively attend to behavioral cues and reinforces the broader thesis that identifying and leveraging such human-aligned data streams is essential for building more general and collaborative AI systems.

3.4.7.1 Future Work

Our findings underpin two key future research directions. First, we are interested in investigating the potential for enhancing the adaptability and personalization of autonomous agents by conditioning them on the information collected about their human teammates. Second, we intend to apply and extend these findings to real-world human-robot collaboration. While we are confident that our general conclusions will extend to these practical scenarios, a real-world domain raises number of interesting questions. These include how to adapt the systems to account for the movement of the human, how to classify the completion of a human action, and determining an appropriate frequency to use when delineating timesteps. Additionally, one potential limitation of this study is that our data was drawn from a participant pool relatively lacking in terms of age, ethnic and cultural diversity. Considering evidence for the culturally contingent nature of gaze patterns (e.g., [420]), future work should explore the cultural nuances of eye gaze as a communicative signal. This is particularly relevant in diverse and multicultural settings where human-robot interactions may be influenced by varying interpretations of gaze behavior.

Chapter 4

Generalization through Structures

In this chapter, I investigate how the use of structure supports generalization. As introduced in Section 2.2, structure refers to the organization of how information flows through a system. While many deep learning systems rely on architectural inductive biases to implicitly structure learning, humans leverage explicit conceptual structures—such as taxonomies, frameworks, and hierarchies—to flexibly generalize and adapt to new settings. These structures enable efficient knowledge reuse, reduce the cognitive and computational search space, and clarify relationships between concepts.

This role of structure is especially critical in the domain of zero-shot coordination, also known as ad hoc teaming, in which agents must collaborate with previously unseen teammates without any prior coordination or shared training. Generalization in this context is not merely beneficial—it is a prerequisite for success. Human teamwork in such conditions relies heavily on shared task abstractions [348] to align behavior, coordinate roles, and adapt to dynamic contexts [401].

Motivated by these findings, I introduce Hierarchical Ad Hoc Agents (HA^2), which leverage a human-interpretable hierarchical task structure to support coordination in novel teaming scenarios. **Empirical results, corroborated by user studies, show that these explicit hierarchical structures significantly enhance an agent's ability to generalize to both new partners and shifts in the environment, ultimately enabling more effective and human-like collaboration.**

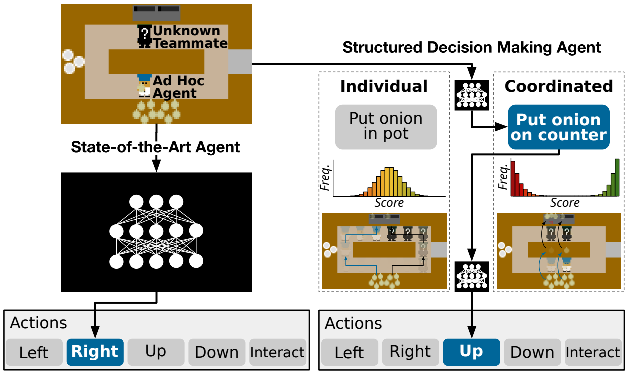


Figure 4.1: Depicted is a scenario in the Overcooked game where an agent is working with a new teammate. The agent could choose to play a coordinated strategy that is more efficient than an alternative individual strategy, but runs the risk of failure if cooperation is not achieved. Successful ad hoc teaming requires not only being able to perform multiple strategies, but also know when to apply the different strategies. Current SotA approaches subsume these decisions into a single black-box; in contrast, we propose that a structured approach to these decisions provides significant benefits.

4.1 Implicitly Aligning Humans and Autonomous Agents through Shared Task Abstractions

The work described in this section has been published as an extended abstract in AAMAS 2023[13], and as a full paper in IJCAI 2025 [17].

4.1.1 Introduction

Successful collaboration requires individuals to efficiently adapt to new teammates. This capability, often referred to as ad hoc teaming [24] or zero-shot coordination [152], is an area where humans consistently outperform state-of-the-art autonomous agents. We argue that this disparity arises because humans have access to shared task abstractions [348], which provide a common foundation that facilitates seamless, implicit coordination. In this paper, we argue that maximizing an agent's ability to collaborate with humans requires providing them with shared task structures and demonstrate the effectiveness of this approach through a large-scale human study.

To elucidate the intricacies and challenges of zero-shot coordination between humans and agents, let's analyze a scenario in the collaborative game "Overcooked", in which players serve as many soups as possible within a time limit. In Figure 4.1, the ad hoc agent is playing with an unfamiliar teammate and must decide how to act next. One option, which we define as the '*individual*' strategy, involves obtaining an onion and placing it directly into the pot. This behavior is conservative but suboptimal: it can achieve a moderate score with any player, but will never achieve a top score. Conversely, the agent could opt for a '*coordinated*' strategy, where the blue agent passes onions on the middle counter, hoping their teammate moves them from the counter to a pot. This strategy is more efficient since it eliminates the long walk around the kitchen, but it carries risk as success hinges on both agents adhering to the strategy. This example illustrates a key challenge in zero-shot teaming: an agent must not only *acquire multiple distinct behaviors*, it also needs to *be able to quickly identify which behavior is most suitable for its teammate's skill level.*

Recent works have tried to overcome the challenges of rapidly adapting to new teammates by leveraging teammates of diverse capabilities. Agents have been trained with teammates that emulate human behavior [57] or with a population of teammates that varying levels of proficiency [354, 231, 227, 210]. However, these approaches sidestep a key factor for achieving genuinely collaborative interaction; as depicted in Figure 4.1, current state-of-the-art systems consolidate high-level strategy decisions and low-level movement decision into one black box model. In contrast, humans are known to leverage hierarchical frameworks for cognitive processing [401], task management [10, 431], and human-human coordination [348], and human-robot collaboration [314, 238]. Further, it has been proposed that structured hierarchies are a core component of the human ability for fast generalization [372]. In this paper, we design autonomous agents equipped with hierarchical structures that provide shared task abstractions that enable more efficient alignment with humans. These structures enable agents to focus on the most relevant information for the respective level of abstraction, prevent agents from overfitting to specific training patterns, and create task-oriented agents who may be more understandable to humans. While the benefits of shared task hierarchies are well-established in certain research domains [161, 390, 10], state-of-the-art methods in human-agent interaction [227, 210] have yet to capitalize on this critical concept. This paper addresses this significant gap and advocates that shared task hierarchies should play a central role in human-agent interaction. Our findings show that leveraging shared task hierarchies can provide greater improvements compared to increasing diversity of training agents (cf. Section 4.1.5.3).

In all, we present *Hierarchical Ad Hoc Agents* (*HA*^{2}), a method that leverages hierarchical reinforcement learning (HRL) to equip an agent with both low-level, efficient maneuvering behaviors and high-level, team-oriented strategies for effective synchronization with human teammates (cf. Figure 4.2). Importantly, HA^2 is agnostic to the underlying training algorithms, and serves as an augmentative layer that complements state-of-the-art methods (SotA, e.g., [354, 57, 227, 210]), leading to statistically significant improvements. Further, this is a deeply generalizable method, as humans have demonstrated the ability to create tasks hierarchies across a broad range of human-human collaborative tasks [348]. Through extensive evaluations, we find that HA^2 offers statistically significant advantages with highlighted by the following contributions: 1) HA^2 outperforms all baselines by over 18.0% when paired with a set of unseen agents, and 2) by over 18.3% when paired with humans. Moreover, 3) HA^2 is significantly preferred by humans, and found to be more fluent, trusted, and cooperative than baselines. To further test the generalizability provided by hierarchical structures, we test the agents zero-shot on modified versions of the game layouts and show that 4) HA^2 is more robust environmental changes, outperforming baselines by more than 10.5x on these layouts. Code is available at `https://github.com/HIRO-group/HA2`.

4.1.2 Related Works

Zero-Shot Coordination The fast-evolving landscape of deep RL agents that interact in the real world has prompted increased investigation into how they can and should interact with humans [80, 253]. A critical challenge is the development of agents capable of zero-shot coordination with human partners [350].

Training partners Prior research identified the limitations of agents trained via self-play—most notably, their behavioral rigidity. To address this, work has enhanced self-play through robust strategy discovery [152, 78, 326], off-belief learning [151], or training with a population of pretrained agents [241]. Notable advances were made with the use of teammates trained via imitation learning [57], that vary in ability [354], or are specifically trained to be diverse [231, 426, 232]. More recent work has investigated ensembling training partners to create a richer diversity without additional computational costs [227]. Our work is directly compatible with this body of work by augmenting the agents with human-aligned structures.

Intention Prediction and Planners A different line of research has focused on modeling the teammate's intention prediction for collaborative tasks [248, 260]. This work often leverages online planners, which have nice properties for ad hoc agents within certain restrictions [404]. Although [405, 289] employ hierarchical structures in their planners, they lack real-world applicability due to computational constraints for complex environments and have not been tested with real human game-play. [57] compared their models to planning-based methods, but were only able to use planners in two of their five layouts. When playing any agent for which they did not have an accurate model of (e.g., a human), performance dropped dramatically. Though out-of-scope here, we believe that modeling would compliment our proposed method.

Type-based Agents Ad hoc teaming has also been investigated by using type-based agents that rely on a pre-generated population of diverse teammates. The PLASTIC framework [24] offers two strategies: PLASTIC-Model, which employs the most human-like teammate for action planning, and PLASTIC-Policy, which first learns then selects the most appropriate complementary policy for each teammate. This latter approach is paralleled in [204], albeit with a distinct similarity metric. Finally, [64] takes this further by subsuming teammates into world models, and using them to learn respective policies.

Hierarchical Reinforcement Learning As breaking down complex tasks into sub-tasks is used in many facets of life, HRL is a well-studied area [358, 86, 82, 384]. HRL has also been extended to multi-agent cooperation, either by deploying a central manager to oversee multiple agents [4], or by imbuing each agent with its own hierarchical architecture [118, 237]. Other work has ventured into learning the sub-tasks [409, 392]. Most similar to our work is HiPT [225]. However, the work differs in several critical ways: 1) Our motivation stems from aligning structures between humans and agents; thus, our abstracted layer between Worker and Manager is fully human interpretable. 2) Our method consistently outperform HiPT across all layouts. See Section 4.1.5.3 for comparative results. 3) We show that our architecture enables greater generalizeability to shifts in the game layouts, a feature not shown in HiPT. 4) We show that HA^2 provides significant benefits regardless of which training teammates are used.

4.1.3 Method

4.1.3.1 Environment

Following prior work in zero-shot human-AI teaming [57, 354, 13], we study the use of hierarchical structures using all five layouts in the Overcooked environment developed by [57]. The goal of this collaborative game is to serve as many soups as possible in the time limit. To accomplish this, players must perform a sequence of task from retrieving onions and placing them in a pot to serving completed soups. Upon service, the team is rewarded with 20 points.

At each timestep, each player can choose to move {*up, down, left, right*}, *interact* with an object (for picking up/placing/serving objects), or *stay* still. To effectively play Overcooked, agents must both coordinate on high-level sub-tasks and low-level movement patterns. At the sub-task level, players should avoid redundant and inefficient sub-tasks such as each retrieving a dish if only one soup is cooking. At a low-level, players must be cautious to avoid collisions. This layered complexity makes Overcooked a particularly apt testbed for human-agent collaboration.

4.1.3.2 Sub-tasks

In human-human game-play, synchronization typically occurs at this sub-task level. In the Overcooked environment, sub-task identification is facilitated by the *interact* action, which serves as a delineating event. Utilizing it, we enumerate all possible outcomes resulting from the 'interact' action to define our set of sub-tasks: (1) Pick up onion from onion dispenser (2) Pick up onion from counter (3) Pick up dish from dish dispenser (4) Pick up dish from counter (5) Place onion in pot (6) Place onion on counter (7) Get soup from pot (8) Place dish on counter (9) Get soup from pot (10) Place soup on counter (11) Serve soup (12) Unknown

4.1.3.3 HA^2: Hierarchical Ad Hoc Agent

Inspired by the notion that cognitive and behavioral alignment between humans and agents enhances human-AI Teaming, we adapt FuN [384] to introduce HA^2: Hierarchical Ad Hoc Agents. HA^2 aims to

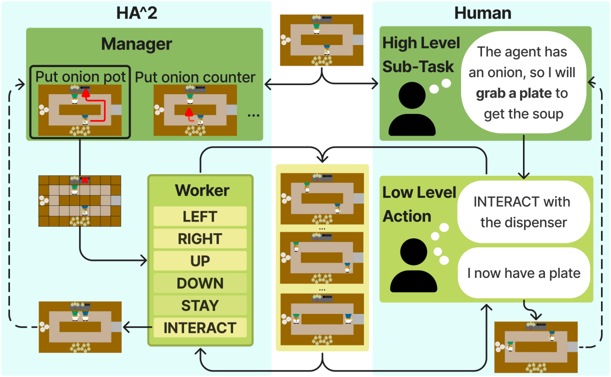


Figure 4.2: An overview of the HA^2 architecture. Similar to human behavior, an observation is initially processed by the Manager to decide on the next high-level sub-task. Subsequently, the Worker executes the necessary low-level actions to complete the sub-task.

approach human-agent teaming as one would approach human-human teaming: by developing a shared and mutually understandable task hierarchy. HA^2 (cf. Figure 4.2) consists of two tiers of models: a Worker, that focuses on the efficient execution of sub-tasks while avoiding collisions; and a Manager that focuses on high-level-task synchronization with its teammate. This decoupled architecture not only facilitates collaboration by allowing the models to focus on the information at their level of abstraction, but it additionally streamlines both their learning processes.

The Worker is tasked with learning how to complete sub-tasks. With reference to Section 4.1.3.2, this consists of moving to a certain location with a specific orientation and *interact*ing with the environment. With this in mind, we add a layer to the lossless observation developed by [57] that indicates the end locations of the current sub-task. For example, for the sub-task ‘put onion in pot’, each non-full pot would be marked in this layer of the observation. We then create a modified versions of the original environment. In this environment, each episode is associated with a sub-task and runs until the agent performs the *interact* action or times out. If the agent completes the correct sub-task, they receive a reward of +1, otherwise they receive a reward of -1. Certain sub-tasks offer additional small rewards for more optimal completion methods. For tasks involving placing objects on or picking up object from counters, an additional reward is added for the numbers of steps that can be saved by using that counter compared to moving from the agent’s current location. For placing onions in pots there is an additional reward for placing onions in the pot that has more onions. When an episode ends, a new sub-task is sampled from the list of possible sub-tasks given the current state. The sub-tasks are sampled inversely proportionally to how often that sub-task has been used previously in training to get a more even coverage of sub-tasks during training. Once the horizon of the original environment is reached, the environment resets the state to the standard start state and a new sub-task is sampled. Due to its undefined nature, we omit the unknown sub-task at this stage.

The Manager is responsible for deciding which sub-task should be completed next. Specifically, it is trained to output a distribution over sub-tasks. To train the Manager, we again create a modified version of the original environment. In this environment, the action space is one of the 12 possible sub-tasks. If the manager selects the undefined sub-task, the additional observation layer passed to the worker is empty. Unlike the base environment, not all actions are possible for each state: for example, the agent cannot put an onion in the pot if they are not carrying an onion. To address this, we mask out all sub-tasks which are not possible at the current time-step. Once the sub-task has been chosen, the associated observation is passed to the Worker, who selects the low-level action for that timestep. We found that having the Manager select sub-tasks at each time-step improved sample efficiency and overall performance given the computational budget. The reward structure is the same as the base environment with a reward of 20 for each soup served.

4.1.4 Experimental Design

4.1.4.1 Baselines and HA^2 models

We implement two baselines representative of the existing approaches in the field: Behavioral Cloning Play (BCP), [57] and Fictitious Co-Play (FCP) [354]. BCP[1] was designed to have an RL agent learn how to play with the movement patterns of a human. To do this, a behavioral cloning (BC) model is first trained from human data, and then a RL agent is trained with the BC model as its teammate. FCP is designed to have agents learn to play with a wide range of teammates. It first learns a population of self-play agents who vary in architecture and seed. It then augments the population by using three versions of each agent: its random initialization, roughly midway through training, and after completing training. It then trains the FCP agent to play with the whole population of agents.

---

[1] BCP was originally named PPO$_{BC}$ in [57] and renamed by [354] to BCP. We use BCP in this paper for succinctness.

We note that HA^2 serves as an architectural enhancement within the agent, and that the agent can be trained using any type of teammate. To demonstrate the applicability of HA^2 to existing methods, we train two versions of HA^2. HA$^2{}_{BCP}$ is trained using a BC teammate and is directly comparable to BCP. HA$^2{}_{FCP}$ is trained using a FCP population and is directly comparable to FCP. We train five iterations of each of the four agents using different random seeds and report the mean and standard error across seeds.

To train the BC models, we closely follow the implementation in [57], using their feature encoding as observation as well as their provided data. We make two small changes which we found improves performance. First, we remove all time-steps where both agents perform the *stay* action. Second, in the loss, we weigh each action inversely proportional to their frequency in the dataset. Following [57], We divide the data in half, and train two models. The better model is used as the human proxy, and the worse model as the BC model. We note that these two agents are the only agents where we train one model per layout.

The RL agents train one model for all layouts and use the 7x7 egocentric view developed by [354]. However, instead of the convolutional neural network (CNN) used in [354], we flatten the observation and pass it through a two-layer multilayer perceptron (MLP) as we found it outperforms a CNN. We experiment using recurrent networks, as in [354], but found they also underperform MLPs. We additionally experiment with frame stacking, which we found outperforms a Recurrent PPO, but underperforms the standard PPO approach.

The training population for the FCP agent and HA$^2{}_{FCP}$ consists of eight self-play agents that vary in seed, hidden dimension (64 and 256), and whether or not they use frame stacking. When training the population, we found that agents learned on the different layouts at different rates. To maintain a good balance of skill levels for each layout, we use different middle checkpoints for each population agent for each layout, with the checkpoints corresponding to points closest to where the agent reaches half the highest score for that layout.

Each population agent was trained for 10 million in-game steps and the BC agents were trained for 300 epochs. HA^2 and the baselines train at different rates with HA^2 taking the longest to train since it requires two predictions — one from the manager, the other from the worker — at each timestep. To keep a fair comparison, we train each agent for 48 hours using the same V100 GPU. For HA^2, we use 24 hours for the

Worker and 24 hours for the Manager. The 48 hours equate to $\sim$119 million timesteps for BCP, $\sim$119 million timesteps for FCP, and $\sim$66 million timesteps for HA^2 (31 million for the Worker and 35 million for the Manager). We note that all agents reached over 98% of their top performance within the first half of this training.

4.1.4.2 Research Questions and Experiments

RQ1: Does HA^2 improve performance with unseen agents? We hypothesize that the addition of a hierarchical structure will help the agent's models focus more closely to the salient information at their respective level of abstraction. Further, we hypothesize that it learn more general game concepts by preventing it from over-fitting to any specific training patterns. Since the reward is fully shared and because the agent can impact its training teammate's actions by influencing the observations, it follows that the agent will also maximize its actions to promote its teammates' high-scoring behaviors. When using low-level actions, this can quickly lead to weird specificities that generalize poorly—e.g. waiting to put the onion in the pot until the teammate is in a specific spot and facing a specific way. Enforcing a hierarchical structure should mitigates this effect since the Worker is not rewarded by teammate behaviors and the Manager has no control over the movement of the Worker.

To test this initial hypothesis, we compare the performance of HA$^2{}_{BCP}$ and HA$^2{}_{FCP}$ to their respective baselines when paired with three agents of varying capability: a self-play model (fully distinct from any in the FCP population), the human proxy model, and an agent that performs random actions. See Section 4.1.5.1 for the results of this experiment.

RQ2: Does HA^2 create higher performing and more fluent human-agent teams? The primary motivation for this work is to develop agents that are effective at collaborating with humans. Human teammates present unique challenges to autonomous agents—prime among them the fact that humans have a significantly higher ability to adapt. In turn, this requires agents paired with humans to not only being able to adapt themselves, but also to make it easy for a teammate to adapt to them. Beyond the improved generalizability we test for, we hypothesize that HA^2's structure will make them more task-focused and in turn more understandable to humans.

To this end, we conduct an IRB-approved online user study. We use a within-subjects design for the study where each participant plays with two agents on each layout. To test our above hypothesis, we evaluate both objective performance and subjective preferences between pairs of agents. Each participant was first provided with an instruction page, before completing a short tutorial that required them to complete all the steps to serve a soup to move on. Each participant then played an 80 second round (400 steps at 5FPS) with each agent on one of the layouts. Between each round, the participants had to answer eight questions adapted from [143] asking them how much they agreed or disagreed with statement on a 7-point likert scale. After each pair, they were asked to rank which of the two agents they preferred. They then repeated this process for the other four layouts. The order of the layouts and agent they played with first within each layout was randomized. The chef that the agent and human controlled were consistent between the two comparative agents, but randomized between layouts and participants.

For this research question, we run two pairwise comparisons: $\text{HA}^2{}_{BCP}$ vs. BCP and $\text{HA}^2{}_{FCP}$ vs. FCP. We recruit 50 participants for the BCP comparison and 25 participants for FCP comparison. We filter out any participant that did not complete the full trial. We additionally filter out any pair of rounds (i.e., comparing two agents on one layout) where the human performed fewer than five subtasks in either round. This leaves us with 47 and 24 participants respectively. We recruit all participants from prolific.com. Participants were compensated using a base rate of \$3.00 plus a bonus incentive of \$0.04 for each dish served. The average participant compensation for these two studies was \$15.79/hour. This human survey was approved an Institutional Review Board, indicating that it presented minimal risk to participants. All participants provided informed consent for the study. Results for these human studies are in Section 4.1.5.2.

RQ3: Can HA^2 agents generalize better to changes in the layouts? Since the hierarchical structure we are using is intrinsic to Overcooked at large, we posit that HA^2 should not only generalize better to different agents, but also generalize better to shifts in the layouts. For this experiment, we create a modified version of each layout by swapping two tiles in each layout. Since we do not have any trained unseen agents on these layouts, we evaluate the HA^{2}s and their respective baselines on these modified layouts by teaming each agent with themselves. Section 4.1.5.4 shows results for this experiment.

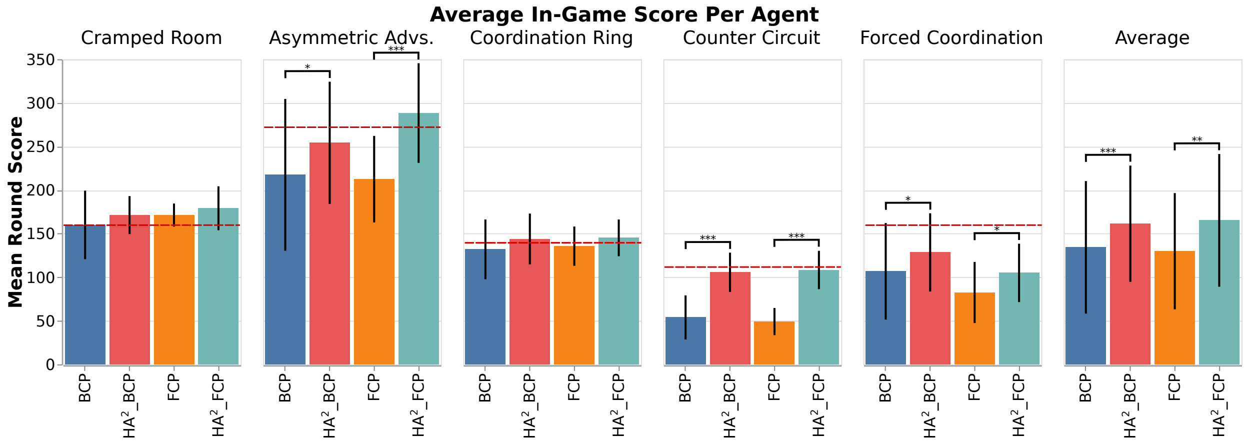


Figure 4.3: Average score of HA^2s and the BCP and FCP baselines when paired with humans on each of the layouts. Each round was 80 seconds long at 5 FPS (T=400 steps). Significance markers: $*=p< 0.05$, $**=p< 0.005$, $***=p< 0.0005$. The red line indicates the max human-human score achieved on that layout from [57] normalized to 400 steps.

4.1.4.3 Significance Testing

For each pairwise comparison, we perform t-tests to measure significance. For the significance of team performance, we compare the score achieved directly. For the ranking significance, we mapped every instance where an agent was preferred over its counterpart to a score of 1 and every other instance to a score of 0. We then used these scores to perform the t-tests. For the Likert questions, we mapped each agreement level to a score between -3 (strong disagree) and 3 (strong agree), with the neutral score being 0. We normalize all participants scores to have a mean of 0 and then use these score for perform the t-tests.

	BCP	HA$^2{}_{BCP}$	FCP	HA$^2{}_{FCP}$
AA	$199.9_{\pm 8.0}$	$278.3_{\pm 6.3}$	$210.8_{\pm 40.0}$	$\mathbf{293.5}_{\pm 7.2}$
CoR	$79.2_{\pm 4.2}$	$133.3_{\pm 3.2}$	$138.6_{\pm 2.5}$	$\mathbf{147.6}_{\pm 0.8}$
CC	$17.1_{\pm 11.4}$	$91.2_{\pm 5.0}$	$74.3_{\pm 19.3}$	$\mathbf{99.9}_{\pm 2.8}$
CrR	$143.1_{\pm 13.8}$	$177.7_{\pm 4.1}$	$183.9_{\pm 4.7}$	$\mathbf{185.5}_{\pm 2.3}$
FC	$73.1_{\pm 5.6}$	$\mathbf{77.6}_{\pm 3.5}$	$56.7_{\pm 4.1}$	$58.4_{\pm 4.8}$
Avg.	$102.5_{\pm 4.5}$	$151.6_{\pm 2.4}$	$133.0_{\pm 8.8}$	$\mathbf{157.0}_{\pm 1.3}$
$\sim$ AA	$23.6_{\pm 41.5}$	$157.2_{\pm 40.4}$	$7.6_{\pm 14.2}$	$\mathbf{208.0}_{\pm 28.1}$
$\sim$ CoR	$11.6_{\pm 11.4}$	$\mathbf{152.8}_{\pm 7.0}$	$22.8_{\pm 6.4}$	$143.2_{\pm 12.6}$
$\sim$ CC	$2.0_{\pm 2.5}$	$70.0_{\pm 15.8}$	$9.2_{\pm 14.5}$	$\mathbf{110.0}_{\pm 35.5}$
$\sim$ CrR	$5.6_{\pm 2.9}$	$\mathbf{162.4}_{\pm 15.2}$	$0.8_{\pm 1.6}$	$154.8_{\pm 36.8}$
$\sim$ FC	$10.4_{\pm 8.9}$	$17.2_{\pm 31.5}$	$3.2_{\pm 3.0}$	$\mathbf{20.8}_{\pm 31.7}$
$\sim$ Avg.	$10.6_{\pm 9.5}$	$111.9_{\pm 13.4}$	$8.7_{\pm 2.4}$	$\mathbf{127.3}_{\pm 7.1}$

Table 4.1: Mean$_{\pm \mathrm{SE}}$ score across 5 random training seeds for HA^{2}s and their respective baselines. The score of each trained agent is the average across 10 trials of T=400 steps with each teammate. In the original layouts, the teammates are an unseen self-play agent, the human proxy, and a random agent. In the modified layouts (denoted with $\sim$), the teammate is a copy of the acting agent.

4.1.5 Results

4.1.5.1 Zero-shot Coordination with Unseen Agents

We first compare HA^2 to the baselines—BCP and FCP—on their ability to generalize to new unseen agents. The results in Table 4.1 clearly demonstrate the improvement provided by the hierarchical structure, with the HA^{2}s outperforming their respective baselines on every layout. Using HA^2 afforded an improvement of 47.9% when using BCP, and an improvement of 18.0% when using FCP. HA$^2{}_{BCP}$ performs best on forced coordination, and HA$^2{}_{FCP}$ performs best on all the other layouts and overall. We discuss a possible cause of this in Section 4.1.5.4. We additionally note that HA^2 is more robust to the random seed than the baselines, with a lower standard error on each layout across the 5 random seeds.

4.1.5.2 Zero-shot Coordination with Humans

We now present the findings of our human study comparing HA^2 and the baselines. Results in Figure 4.3 demonstrate that in both HA$^2{}_{BCP}$ and HA$^2{}_{FCP}$ significantly outperform their respective counterparts on the overall score achieved, and on the asymmetric advantages, counter circuit, and forced coordination layouts. We note that the scores of the baselines in cramped room and coordination ring are closer to the maximum human-human [2] score achieved (dotted red line), leaving less room for improvement. As such, we anticipated there would be a smaller variability on these layouts. Table 4.2 further shows that HA^2 was significantly preferred over their counterparts. In RQ3, we had hypothesized that HA^{2}s would improve human-agent teaming because they are easier to understand, and therefore easier to adapt to. Figure 4.4 supports this hypothesis and shows that in both comparisons of HA^2 to the baselines, humans rated the HA^{2}s as significantly more understandable, intelligent, and cooperative. In the case of FCP and HA$^2{}_{FCP}$, humans also found that HA^2 was significantly more fluent, trusted, and more helpful at helping the humans adapt to the task. These results strongly support using shared task hierarchies for human-agent collaboration.

In line with the results with unseen agents, forced coordination is the one layout where BCP and HA$^2{}_{BCP}$ outperform their FCP counter parts. We hypothesize that this is due to it being the only layout where having an untrained teammate blocks the agent's ability to earn a reward. Since a third of FCP's training population are untrained agents, FCP and HA$^2{}_{FCP}$ effectively lose a third of their training. The results in the appendix of [354] support this hypothesis showing that forced coordination is least benefitted by FCP. This can likely be remedied by excluding the untrained partners in layouts where coordination is required to achieve a non-zero score.

	% Preferred	p-value
HA$^2{}_{BCP}$ over BCP	57.68	0.0070
HA$^2{}_{FCP}$ over FCP	65.25	0.0000018

Table 4.2: Human preference between pairs of agents and their respective significance.

---

[2] From [57]'s data normalized to 400 timesteps.

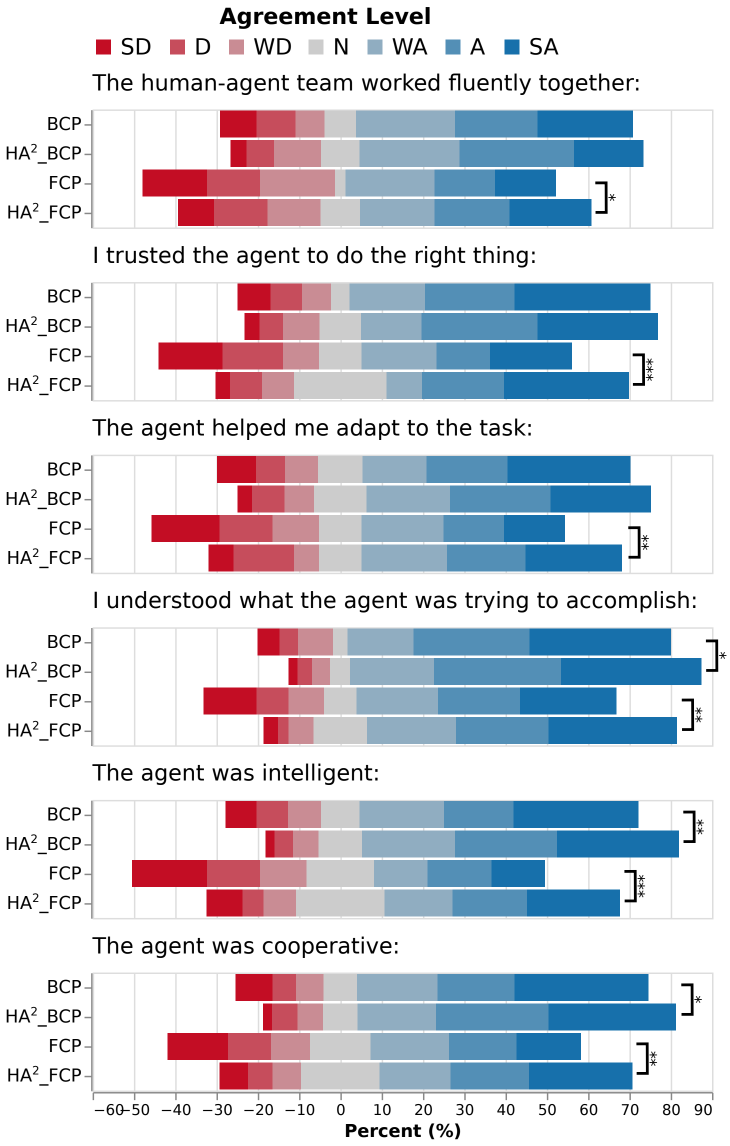


Figure 4.4: Subset of results from the eight Likert-scale questions that participants answer after playing with each agent for the comparison between HA^2 and their baselines. Bars that are more blue indicate that people agree more strongly with the statement. Conversely, more red indicates that people disagreed more strongly with the statement. Significance markers: *=p$<$ 0.05, **=p$<$ 0.005, ***=p$<$ 0.0005. Legend: SD=Strongly Disagree, D=Disagree, WD=Weakly Disagree, N=Neutral, WA=Weakly Agree, A=Agree, SA=Strongly Agree.

Interestingly, when analyzing Figures 4.3 and 4.4, we noticed that even if their overall scores were generally worse, BCP and $\text{HA}^2{}_{BCP}$ were better perceived on every subjective metric relative to their FCP and $\text{HA}^2{}_{FCP}$. This does pose the question of whether utilizing human behavior in training does provide a more human-like game-play, and in turn a more fluid experience for humans, which is supported by the results in [210]. We leave a more thorough investigation of this question, as well as the relationship between team performance and human perception, to future work.

4.1.5.3 Comparison to State-of-the-Art

In Table 4.3, we compare HA^2 to results published in other peer-reviewed work that uses the same overcooked environment. As each method employs a range of design decisions, this table should be viewed as a comparison of systems. Notably, when paired with a human proxy, HA^2 is tied as the best performing agent, whereas when HA^2 is paired with real humans, HA^2 outperforms all other work by more than 23%, showcasing HA^2's adeptness at human collaboration. This is further emphasized when comparing to the most similar work of HiPT. HA^2 outperforms HiPT by 17% when paired with a human proxy, and by 26% when paired with real humans while using 15.1 times fewer timesteps (1 billion timesteps for HiPT vs $\sim$66 million timesteps for HA^2). This highlights HA^2's greatest distinction from HiPT: *using **human-aligned** structures improves the training efficiency and performance of autonomous agents that collaborate with humans.* Lastly, we compare HA^2 to the most recent SotA method: GAMMA [210]. Even with a simpler training population, HA^2 outperforms GAMMA by 25% with a proxy human. Further, when paired with real humans on counter circuit, which is the only original layout on which they provide results with real humans, HA^2 outperforms the best version of GAMMA with a score of 110 compared to 91. In all, we show that shared task structures are a critical component when developing collaborative agents.

4.1.5.4 Generalization to Shifts in Layouts

We report our results on the generalization ability of HA^2 and the baselines on the altered layouts. The latter half of Table 4.1 shows that BCP and FCP overfit to the specific layouts and their performance drops dramatically when the layouts are changed. In contrast, the HA^{2}s are able to maintain a reasonable performance, and are over 10.5x better on the modified layouts. Together with the results in Section 4.1.5.1, these results provide strong support for our hypothesis that the hierarchical structure enables the model to learn more generalizable concepts about collaboration and game-play.

	Training Steps	W. Proxy	W. Humans
FCP	1.0e9	**157**	119
MEP	5.5e7*	98	98
TrajeDi	5.5e7*	76	87
PECAN	NR	105	134
HiPT	1.0e9	134	131
GAMMA	1.5e8	132	NR
$\text{HA}^2{}_{FCP}$	6.6e7	**157**	**165**

Table 4.3: Results comparing HA^2 to other published results. All results are taken from the respective works and adjusted to 400 timesteps, except for TrajeDi's results which are taken from [426]. NR=not reported. * indicates that separate agents are trained for each layout and that the cumulative step count across layouts is presented. FCP [354], MEP [426], TrajeDi [232], PECAN [227], HiPT [225], GAMMA [210],

4.1.6 Discussion

4.1.6.1 Summary

In this work, we investigate how explicitly shared task abstractions can enhance generalization and collaboration in zero-shot coordination settings. Through extensive experiments and user studies, we demonstrate that Hierarchical Ad Hoc Agents (HA^2)—which leverage human-interpretable hierarchical structures—significantly outperform existing baselines across both quantitative performance metrics and qualitative user evaluations. Human participants rated HA^2 as more fluent, understandable, cooperative, and intelligent, and teams paired with HA^2 consistently achieved higher task performance.

Notably, HA^2 exhibits strong generalization across novel agent types and environmental variations, and even when trained with simpler partners, it surpasses all baselines when deployed with real human teammates. These results underscore the critical role of explicit structure in supporting adaptive and robust collaboration. By aligning with the kinds of shared abstractions humans rely on, HA^2 demonstrates how structured representations can serve as a powerful mechanism for enabling more generalizable, resilient, and human-compatible AI systems.

4.1.6.2 Limitations

We now discuss the limitations of our proposed method. The hierarchical structure in HA^2 necessitates additional engineering effort, both in the development of the structure, and the adjustments to the environment required to train the different modules. We note that the method in which to break-down large tasks into sub-tasks to create a hierarchy is not the focus of this work, and has been extensively explored in many domains including human factors research [10, 348], robotics [161], and single-agent long-horizon tasks [390]. Rather, the focus of this work is demonstrating the importance of shared task hierarchies in human-agent collaboration.

4.1.6.3 Future Work

We envision the following avenues for future work:

First, to incorporate explicit mental models of teammate sub-tasks into agent planning, similar to [248, 260]. We envision that these mental models will synergize with the abstracted manager sub-tasks allowing for more efficient computation of these models, and in turn providing the manager with an efficient understanding of human team members' capabilities and intentions.

Second, we believe HA^2 shows promise as a framework to investigate human-agent communication in collaborative games; it is much easier to communicate at sub-task-level than at action-level.

Chapter 5

Generalization by Building Theories

In this chapter, I investigate mechanisms of adaptation, specifically how internal representations can be updated based on limited data. While modern machine learning relies heavily on gradient-based optimization over large datasets, humans are capable of rapidly adapting by forming and testing high-level hypotheses about the causes of observed behavior. This hypothesis-driven learning process enables efficient generalization, especially in low-data regimes.

This capacity is particularly important in the domain of few-shot behavioral cloning, where agents must accurately mimic user behavior based on a small number of demonstrations. Rather than memorizing surface patterns, humans adapt by leveraging prior knowledge and identifying the key factors that differentiate the new situation from familiar ones [344, 116]. In the context of imitating others, this means constructing and refining a mental model of the user, continuously updating it through observation and inference [173, 128].

Inspired by this human approach, I developed PROSE, a method for inferring and refining latent user preferences in order to explain observed writing samples. These preferences are then used to guide a Large Language Model (LLM) to better emulate a user's writing style. This iterative refinement leads to more accurate personalization and robust generalization in settings with minimal data.

5.1 Aligning LLMs by Predicting Preferences from User Writing Samples

The work described in this section has been published in ICML 2025 [14].

5.1.1 Introduction

People increasingly rely on LLM-powered AI Assistants to complete tasks on their behalf, such as creating written materials: “write a professional email about the following great idea” or “summarize this new article for me to share share with my friends”. As the writing style learned by an LLM during pretraining is generic, it may not match the user’s preferred writing style and voice [61, 324], leading to outputs that feel impersonal, misaligned, or requiring extensive editing.

Existing approaches to learn preferences rely on preference rankings (RLHF) [435, 299], demonstrations [270, 332], prompting [430], and user edits [109]. However, methods such as RLHF and SFT (on user demonstrations) require a large number of samples, and do not learn the preferences in a form users can interpret or interact with. In-context learning (ICL) from user demonstrations does learn from a small number of user demonstrations, but lacks interpretability and offers limited control to the user, and prompting approaches require the challenging task of identifying a high-quality prompt [412]. Furthermore, methods that learn from user edits ignore data about user preferences and style that are available from observing how the user completes writing tasks on their own.

[109] introduces CIPHER to establish the benefits of aligning a LLM through prompting by learning a description of user preferences compared to ICL conditioned on user demonstrations (i.e., needing fewer tokens, interpretable representation, and a modifiable representation). The preference description is learned from user edits on the assistant’s generations. However, CIPHER does not enable the LLM to reflect on and refine its inferred preference description, which limits the assistant’s ability to adapt to a user nuanced writing style.

In this paper, we build on CIPHER and introduce **PROSE** (**P**reference **R**easoning by **O**bserving and **S**ynthesizing **E**xamples), a novel approach that leverages two key innovations to enhance the precision and efficacy of the preference description inferred from user demonstrations: (1) iteratively refining the inferred description until the assistant’s generations closely align with the user, and (2) verifying the inferred preferences across multiple user demonstrations. The inferred description is used to condition the LLM to generate writing more aligned with the user.

We evaluate PROSE on PRELUDE [109], the assistive writing benchmark accompanying CIPHER, and identify several limitations. First, the ground truth preference sets often overlap, lack diversity, and match the default LLM behavior. Second, edits are performed only if the assistant generation is inadequate, meaning it is not possible to distinguish between good and excellent generations. Lastly, PRELUDE relies on user edits as the learning signal, meaning the assistant's initial draft can limit the quality of the final writing sample. To address these limitations, we introduce a novel assistive writing benchmark, PLUME (**P**reference **L**earning from **U**ser **E**mails and **M**emos).

We systematically evaluate the benefits of PROSE on PLUME using four LLMs ranging in size and ability, and find that PROSE outperforms CIPHER by 33% [109]. Additionally, we demonstrate PROSE can be combined with ICL to further improve over CIPHER by 47% and up to 9% over ICL. In all, our contributions are:

- PROSE: A new method to infer user preferences.
- PLUME: An improved benchmark for preference inference from user writing demonstrations.
- An in-depth ablation study on PROSE's iterative refinement and consistency verification steps
- An analysis comparing learning explicit preference descriptions and conditioning directly on in-context examples

5.1.2 Related Work

Personalizing LLMs In natural language generation, prompting [297] and in-context learning [50] have proven effective methods for controlling the generation of text, especially in a preference-driven context [357, 356].

Some prior approaches for adapting models to user preferences involve RLHF [349] and fine-tuning [366, 433], which can be compute-intensive and inaccessible to some practitioners without the budget or scale of needed data. To reduce data requirements, [332] propose treating user demonstrations as implicitly preferred over all model outputs, allowing for more efficient preference modeling. Another line of work aims to minimize compute demands by identifying and selectively adjusting internal activations to steer model behavior [205, 376, 217]. While effective for promoting broad, predefined objectives—such as improving truthfulness or reducing toxicity—it remains unclear how such techniques can generalize to individual users without explicit guidance. With the rise of LLMs with strong instruction-following capabilities, methods like prompting to adapt to a user's profile have become more popular [334, 322]; however, these methods too often rely on explicit user feedback to optimize prompts [214]. PROSE circumvents these issues by learning from implicit user signals, breaking down preferences into sub-components to generate tailored user-preferences, all without the need for fine-tuning.

Preference-Conditioned Agents Combined preference inference and conditioning has recently gained traction, with the following three works most aligned with PROSE.

[277] explores preference learning in quadrupedal mobile manipulation using an object detection module to map image observations to text. An LLM then infers preferences by comparing pairs of trajectories. These preferences are in turn used to improve task alignment with user preferences. [333] train a preference inferring model that outputs a set of rules to use during generation, and demonstrate improved personalization on a set of writing tasks. Lastly, [109] propose the PRELUDE environment, where an LLM learns writing style preferences in a collaborative authoring task. We discuss this work in detail in Section 5.1.4.

These methods all rely on a single inference step, whereas our approach uses iterative refinement to learn more precise preferences, and preference verification across several user examples for robustness.

5.1.3 PROSE

PROSE aligns an AI writing assistant with a user's preferences $\bar{\mathbf{p}}_u$ by learning a preference description $\hat{p}_{\text{desc}}$ that allows the assistant (an LLM) to mimic the user's demonstrations $\mathbf{w}_u$, which are determined by $\bar{\mathbf{p}}_u$. For example, learning that articles should be summarized in the style of an old timey radio broadcast.

Each time the user gives the assistant a new task or provides a new task-description and demonstration pair (x_{task}, w_u), following [109] PROSE retrieves up to three previously observed demonstrations relevant to the given task along with the preferences inferred from those demonstrations from its interaction memory. The retrieved preferences are then aggregated to form the preference description $\hat{p}_{\text{desc}}$ using the prompt in Figure D.5 (Appendix D.6.1), which is used to condition the assistant during generation: $w_a = \texttt{generate}(\texttt{llm}, x_{\text{task}}, \hat{p}_{\text{desc}})$. If no demonstrations have been seen, the AI assistant is not conditioned on any preferences, $w_a = \texttt{generate}(\texttt{llm}, x_{\text{task}})$.

If the AI assistant's generation, w_a does not match the user's demonstration, w_u, the inferred preference description $\hat{p}^0_{\text{desc}}$ is updated via **iterative refinement** (Section 5.1.3.1) steps and a **preference consistency verification** (Section 5.1.3.2) step – PROSE's contributions. Iterative refinement alternates between updating the inferred preference description $\hat{p}^{s+1}_{\text{desc}}$ by comparing the agent's generation, w_a, to the user's demonstration, w_u, and rerunning generation conditioned on the updated $\hat{p}^{s+1}_{\text{desc}}$ until either the maximum number of iterative refinement steps (S) is reached or no updates to the inferred preference description are made. Consistency verification breaks the final $\hat{p}_{\text{desc}}$ into preference components and prunes components that are not supported by previously seen demonstrations.

A visualization of PROSE (top) and the prompt summaries (bottom) for each of its preference inference steps are provided in Figure 5.1. The algorithm is provided in Appendix D.1, and the complete prompts are in Figure D.5 (Appendix D.6.1)[1] .

5.1.3.1 Iterative Refinement

To improve $\hat{p}_{\text{desc}}$, the LLM is prompted to compare and contrast w_a and w_u and then modify $\hat{p}_{\text{desc}}$ such that the modification reduces the difference between w_a and w_u: $\hat{p}^{s+1}_{\text{desc}} = \texttt{generate}(\texttt{llm}, x_{\text{update}}, \hat{p}^{s}_{\text{desc}}, w_u, w_a)$, where x_{update} =“Preference Update Prompt” in Figure 5.1. The updated preference description is accumulated in $\hat{\mathbf{p}}_{\text{desc}} = [\hat{p}^0_{\text{desc}}, ..., \hat{p}^s_{\text{desc}}]$, where s is the iterative refinement step.

PROSE then conditions the LLM on the updated preference description $\hat{p}^{s+1}_{\text{desc}}$ to generate a new writing sample w^{s+1}_a. The process of generating AI assistant writing samples, comparing to the user demonstrations, and updating the inferred preferences continues until either the candidate solutions exactly match the user's demonstrations, the preference description is unchanged between subsequent update steps, or a maximum number of iteration steps is reached (S). Qualitative examples of the consistency verification procedure are in Appendix D.6.5.

---

[1] code coming soon!

5.1.3.2 Consistency Verification

After the preference description is improved through iterative refinement, each component of each preference description in $\hat{\mathbf{p}}_{\text{desc}}$ is verified against relevant, previously observed user demonstrations. The verification step removes preference components that were incorrectly inferred or are overly specific to a single demonstration.

Consistency verification operates on the component level (e.g. “use emojis”, “use alliterations”). Therefore, the natural language preference descriptions (e.g. “write a tweet with emojis and alliterations”) produced by iterative refinement are first broken into components by prompting the LLM to convert the preference description into an ordered set of preference components. The preference components are aggregated over all preference descriptions to help avoid over fitting: $\hat{\mathbf{p}}_c = \bigcup_{s=0}^{|\hat{\mathbf{p}}_{\text{desc}}|}(\texttt{generate}(\texttt{llm}, x_{\text{breakdown}}, \hat{p}^s_{\text{desc}}))$, where $x_{\text{breakdown}}$ =“Breakdown Prompt” in Figure 5.1.

PROSE verifies each preference component in $\hat{\mathbf{p}}_c$ against each of the relevant user demonstrations by prompting an LLM to assign a score $v^i_{\text{score}} \in [-2, 2]$ indicating how strongly the demonstration confirms the preference: $v_{\text{score}} = \frac{1}{|\mathbf{w}_u|}\sum_{i=0}^{|\mathbf{w}_u|}(\texttt{generate}(\texttt{llm}, x_{\text{verification}}, \mathbf{w}^i_u, \hat{p}^s_{\text{desc}}))$, where $x_{\text{verification}}$ is “Consistency Verification Prompt” in Figure 5.1. If v_{score} is below the specified threshold (v), the preference component is removed. The task description, user demonstration, and final preference components $(x_{\text{task}}, w_u, \hat{\mathbf{p}}_c)$ are then stored in PROSE’s interactive memory. Qualitative examples of the consistency verification procedure are in Appendix D.6.4.

5.1.4 Assistive Writing Benchmark

5.1.4.1 PRELUDE

[109] propose PRELUDE (**PRE**ference **L**earning from **U**ser’s **D**irect **E**dits) to evaluate algorithms that infer preferences for assistive writing tasks. Success is defined as: (1) maximizing the quality of the inferred user’s preferences and (2) minimizing the amount of work required by a user to edit the generated text into an acceptable form.

PRELUDE consists of two tasks: summarizing articles and writing emails from notes. Each task has a set of users, and each user has distinct preferences per summary and email topic (e.g., summarize an encyclopedia article versus news article). The summarization and email writing tasks have five and four users respectively. See Table D.7 (Appendix D.4) for the mapping between users, topics, and preferences.

To solve a given task, the agent must write a summary or email using the provided article or notes along with any preferences the agent has inferred. The user is then asked if the agent's generation is satisfactory based on the user's true preference. If the agent's generation is satisfactory, the agent accrues no penalty. If the agent's generation is not satisfactory, the user edits the agent's generation, and the agent is penalized based on the extent of the edits. The agent observes the user's edits to improve its inferred preferences.

We analyze PRELUDE and find that the (1) chosen metrics, (2) the editing process, and (3) the ground truth preferences are key limitations of the benchmark, that lead to a weak correlation between the quality of the inferred preferences and the quality of the generated writing.

Metric Correlation As the goal is to infer user preferences, the measure of the agent's generation quality (i.e., the user-edit-based penalty) must be highly correlated the quality of inferred preference. We measure the correlation between PRELUDE's *preference quality metric* — preference accuracy[2] — and *generation quality metric* — Levenshtein distance [201] between the LLM generation and user edited generation. For a each summary and email topic, we generate the powerset of PRELUDE's ground truth preferences and create a population of agents. Each agent is conditioned on a subset from the powerset and completes its assigned task for five seeds. The quality of the inferred preferences and of the resulting generations is measured according to PRELUDE's performance metrics. We calculate the Pearson correlation between each of PRELUDE's *preference quality* and *generation quality* metrics:

$$\rho_{P,G} = \frac{\mathrm{Cov}(P, G)}{\sigma_P \sigma_G}$$

where P denotes the measured preference quality and G denotes the measured generation quality. We report a subset of the results in Table 5.1 (Full results in Appendix D.3.1).

The results, reported in Table 5.1, show a weak correlation (< 0.5) between PRELUDE's preference accuracy and Levenshtein distance metrics. The accuracy metric relies on the "highest" BERTScore, and

---

[2] a preference is correct if its BERTScore [419] with true preference set is greater than the BERTScore with any other preference set.

Metric	PRELUDE Acc.	PRELUDE P. Sim.	PLUME Acc.	PLUME P. Sim.
Levenshtein dist	-0.43	-0.39	0.01	-0.11
PPCM	0.42	**0.42**	0.39	**0.73**

Table 5.1: Subset of Pearson correlation ($\rho_{P,G}$) between preference quality metrics and generation quality metrics across both the summarization and email tasks. Best correlation in each framework is bold. P. Sim. (Preference similarity) and PPCM (Per Preference-Component Match) are described in Section 5.1.4.2. Full results in Appendix D.3.1.

therefore cannot differentiate partially correct preferences from perfectly correct preferences. Moreover, the Levenshtein distance varies substantially between generations even when conditioned on the exact same preferences (an illustrative example is in Appendix D.5.1). [109] allude to this as a motivation for their two-stage editing process, and when we compare the results to a version of PRELUDE where the user always generates summaries or emails directly from the article or notes instead of editing the agent's summary or email ($\text{PRELUDE}_{\text{NoEdit}}$), we see a further drop in correlation. However, we propose addressing this issue using improved metrics.

The Editing Procedure Relying on a binary label to indicate whether a generation matches the user's preferences is inherently ambiguous. It is not possible to distinguish between generations that align with 65% versus 100% of preferences. Even if this ambiguity is resolved, generations not selected for editing incur no cost and provide no incentive to further improve the quality of the inferred preferences. Lastly, the editing process unduly influences the user's writing, as demonstrated in Appendix D.5.2.

Preference Sets We observe the following limitations with PRELUDE's preference sets: (1) certain preference components have minimal impact on the generated text, due to unclear definitions (e.g., "skillful foreshadowing") or similarity to default LLM behavior (e.g., "clear"); (2) preferences are repeated across several task topics (e.g., "short", "brief", "concise" appear in four of five summarization preference sets); and (3) there is a large variance in preference set complexities across users (e.g., "targeted to young children, storytelling, short sentences, playful language, interactive, positive" vs."question answering style"). PRELUDE's preferences are in Appendix D.4 (Table D.7)

Knowledge of Topics Instead of treating each task topic as a distinct user, PRELUDE introduces the additional challenge of context awareness; each user has different preferences based on the task's topic.

Therefore, prior to writing a summary or an email the agent must first identify the correct context, an orthogonal challenge to inferring preferences.

5.1.4.2 PLUME

To address PRELUDE's limitations, we develop a new environment PLUME (**P**reference **L**earning from **U**ser **M**emos and **E**mails) based on same underlying tasks and topics as PRELUDE. As in [109], PLUME uses `GPT-4o` as a proxy human user. In the following sections, we provide a detailed description of how PLUME addresses each of PRELUDE's limitation.

Metric Correlation We investigate and compare new preference and generation-quality metrics. For the *preference quality metric*, we evaluate an LLM-as-a-Judge [427] metric that prompts an LLM to identify how similar the inferred preference description is to the true preference description on a 5-point Likert scale, which we call Preference-Similarity. For the *generation quality metric*, we evaluate length-normalized Levenshtein distance (ln-L-dist), BERTScore, and an LLM-as-a-Judge [427] metric inspired from the editing procedure in PRELUDE. The LLM-as-a-Judge evaluation is a per preference-component match (PPCM) that asks an LLM how much a component of a the ground truth preference is exhibited in a piece of writing on a five point Likert scale from "clearly contradicts" (score of -2) to "clearly exhibits" (score of +2). This is repeated for each component of the true preference set, and we compute the mean score across components. The full prompts used for both of the LLM-as-a-Judge metrics are shown in Appendix D.2 (Figure D.1 and Figure D.2).

The results in Table D.1 (Appendix D.3.1) show that Preference-Similarity has a stronger correlation with each writing generation metric than PRELUDE's accuracy metric. Looking at the generation quality metrics, Levenshtein distance consistently has the weakest correlation and PPCM the strongest. Notably, the pairing of Preference-Similarity (preference quality) and PPCM (generation quality) provides the highest correlation in every situation and are the primary metrics we report in PLUME.

The Editing Procedure In place of the editing, PLUME has the agent and user independently solve each task to (1) enable the agent to learn from every user example, unless the agent's generation exactly matches the user's; (2) remove ambiguity about whether a generation should be edited and incur a cost;

(3) provides a smoother curve along which to evaluate different methods; and (4) prevents agents from influencing users.

	Summarization		Emails		Tasks Mean	
Method	Pref. Sim.	PPCM	Pref. Sim.	PPCM	Pref. Sim.	PPCM
No Learning Baselines						
NPC	$0.00_{\pm 0.00}$	$-1.09_{\pm 0.03}$	$0.00_{\pm 0.00}$	$-0.91_{\pm 0.03}$	$0.00_{\pm 0.00}$	$-1.00_{\pm 0.02}$
Oracle	$3.86_{\pm 0.07}$	$1.71_{\pm 0.04}$	$3.89_{\pm 0.06}$	$1.95_{\pm 0.01}$	$3.87_{\pm 0.05}$	$1.83_{\pm 0.02}$
Learning Baselines						
ICL	$0.00_{\pm 0.00}$	$\mathbf{1.35}_{\pm \mathbf{0.08}}$	$0.00_{\pm 0.00}$	$\underline{1.39_{\pm 0.07}}$	$0.00_{\pm 0.00}$	$\underline{1.37_{\pm 0.05}}$
CIPHER-1	$1.21_{\pm 0.04}$	$-0.05_{\pm 0.06}$	$\underline{1.67_{\pm 0.07}}$	$0.33_{\pm 0.05}$	$1.44_{\pm 0.04}$	$0.14_{\pm 0.04}$
CIPHER-5	$1.24_{\pm 0.07}$	$-0.08_{\pm 0.09}$	$\mathbf{1.69}_{\pm \mathbf{0.07}}$	$0.25_{\pm 0.07}$	$\underline{1.46_{\pm 0.05}}$	$0.09_{\pm 0.06}$
PROSE Ablations						
PROSE_{CE}	$1.23_{\pm 0.06}$	$0.51_{\pm 0.08}$	$1.46_{\pm 0.07}$	$0.97_{\pm 0.08}$	$1.34_{\pm 0.05}$	$0.74_{\pm 0.06}$
PROSE_{u}	$1.30_{\pm 0.11}$	$0.47_{\pm 0.10}$	$1.34_{\pm 0.10}$	$0.84_{\pm 0.11}$	$1.32_{\pm 0.07}$	$0.65_{\pm 0.07}$
$\text{PROSE}_{u,a}$	$1.35_{\pm 0.10}$	$0.49_{\pm 0.11}$	$1.58_{\pm 0.09}$	$1.04_{\pm 0.06}$	$1.47_{\pm 0.07}$	$0.76_{\pm 0.06}$
$\text{PROSE}_{u,a,S>1}$	$1.37_{\pm 0.11}$	$0.75_{\pm 0.09}$	$1.50_{\pm 0.08}$	$1.21_{\pm 0.08}$	$1.43_{\pm 0.07}$	$0.98_{\pm 0.06}$
PROSE_{NV}	$\underline{1.47_{\pm 0.06}}$	$0.87_{\pm 0.10}$	$1.38_{\pm 0.10}$	$1.18_{\pm 0.08}$	$1.43_{\pm 0.06}$	$1.02_{\pm 0.06}$
$\text{PROSE}_{\text{Full}}$	$\mathbf{1.51}_{\pm \mathbf{0.09}}$	$0.90_{\pm 0.07}$	$1.47_{\pm 0.08}$	$1.24_{\pm 0.07}$	$\mathbf{1.49}_{\pm \mathbf{0.06}}$	$1.07_{\pm 0.05}$
$\text{PROSE}_{\text{Full+ICL}}$	$\underline{1.34_{\pm 0.09}}$	$\underline{1.34_{\pm 0.07}}$	$1.39_{\pm 0.09}$	$\mathbf{1.65}_{\pm \mathbf{0.05}}$	$1.37_{\pm 0.06}$	$\mathbf{1.49}_{\pm \mathbf{0.04}}$

Table 5.2: PROSE's performance on the two tasks measured by the quality of inferred preferences (Pref. Sim.) and preference compliance (PPCM) compared against no preference conditioning (NPC), true preference generation (Oracle), in-context learning (ICL), CIPHER [109], and ablations over PROSE's components. Results are the mean and pooled standard error across the four LLMs and five seeds. Best results are bolded, second best are underlined.

Preference Sets PLUME reworks the preferences according to the following criteria: (1) each preference set contains an equal number of components; (2) within each task, preference sets have a shared structure; (3) as much as possible, preferences components are orthogonal to each other, avoiding overlapping preferences (e.g., "write in the style of old-timey radio" and "use archaic language") or contradictory preferences (e.g., "use emojis" and "use a formal tone"); and (4) preferences components do not follow the LLMs default behavior — i.e., generating an output conditioned on no preference should lead to a lower score than when generating on the preference component. PLUME's preferences are in Appendix D.4 (Table D.7). We encourage future researchers to use PLUME with different preference sets to adjust difficulty or examine specific concepts.

Knowledge of Topics As this work focuses on how to infer preferences, the version of PLUME used in all experiments assumes a distinct known user per topic. We note that PLUME is easily adaptable to use hidden topics if desired.

5.1.5 Experimental Set Up

All experiments consist of three phases. First, the user provides a demonstration using their true preferences. Second, the agent completes the user's task using its currently inferred preferences (if any). Finally, the agent compares its generation with the user's example to infer new preferences to use going forward.

All AI assistants are evaluated on their ability to complete email writing and article summarization tasks on behalf of the user. Each task has different types (e.g., email to your boss versus email to a family member), and each user's preferences differ based on the task type. The assistants are evaluated along two dimensions: *preference quality* to measure the similarity between true and inferred preferences (see Appendix D.2.1), and *generation quality* to evaluate how well an agent's writing aligns with the user's true preferences (see Appendix D.2.2). Both performance measures use LLM-as-a-Judge to assess the similarity between the true and inferred preferences, and between the true preferences and the agent's generations.

The agent aligns itself with four (email) or five (summarization) users with five demonstrations per user. Performance is evaluated per task as the mean across all demonstrations, users, and task type. Each task is run over five seeds (standard error is reported over the seeds). The ground truth user preferences by task and task type are in Appendix D.4 (Table D.7). The performance of four LLMs is reported and compared: `Qwen2.5-7B-Instruct`, `Qwen2.5-72B-Instruct`, `GPT-4o-mini`, and `GPT-4o`[408, 267]. For all LLMs, S and v are determined via a hyper-parameter sweep over $v \in 0, 0.25, 0.5, 0.75, 1$ and $S \in 2, 3, 4, 5$. In our experiments $S = 5$ for all LLMs, and $v = 0.25$ for `Qwen2.5-7B-Instruct`, $v = 0.5$ for both `GPT-4o`models, and $v = 0.75$ for `Qwen2.5-72B-Instruct`. For all experiments GPT-4o is used as a synthetic human. The synthetic human prompts can be found in Appendix D.6.2.

5.1.5.1 Research Questions

RQ1: Does iterative refinement improve performance?

We consider three variants of PROSE: (1) PROSE$_u$ infers $\hat{p}_{\text{desc}}$ given only the user's demonstration w_u;

(2) PROSE$_{u,a}$ infers $\hat{p}_{\text{desc}}$ given the user's demonstration w_u and the initial assistant generation $w_a^{s=0}$; and (3) PROSE$_{u,a,S>1}$ refines $\hat{p}_{\text{desc}}$ over $\leq S$ inference steps given the user's demonstration w_u and the initial assistant generation $w_a^{s=0}$. PROSE$_{Full}$ refines $\hat{p}_{\text{desc}}$ over $\leq S$ inference steps given the user's demonstration w_u and the iteratively refined assistant generations $w_a^{s\in[0,S]}$. Comparing PROSE$_u$ and PROSE$_{u,a}$ measures the effect of comparing assistant generations to the user demonstration when inferring preferences. The differences between the PROSE$_{u,a}$ and PROSE$_{u,a,S>1}$ quantifies the role of increasing the number of refinement steps. Lastly, comparing PROSE$_{u,a,S>1}$ and the the complete PROSE algorithm PROSE$_{\text{Full}}$ clarifies the effects of comparing the user demonstration to the assistant's generation conditioned on the latest inferred preference description.

RQ2: Does filtering preferences that are not relevant to multiple user demonstrations improve performance?

To answer this question, we evaluate a variant, PROSE$_{\text{NV}}$, that does not use the preference consistency verification step Section 5.1.3.2.

RQ3: Is conditioning on preferences better than conditioning on demonstrations?

To answer this question, we compare PROSE, CIPHER [109], and ICL on all tasks, task types, and user profiles. We additionally combine the PROSE and ICL to measure the extent to which they are complementary.

5.1.5.2 Baselines

In addition to the PROSE baselines outlined Section 5.1.5.1, we implement the following models.

We implement CIPHER-1 and CIPHER-5 [109], and an in-context learning (ICL) agent using previously observed user demonstrations. The CIPHER baselines are adapted to learn from PLUME's user demonstrations instead of user edits.

We then implement three additional baselines. An agent that solves the task with no preference conditioning (NPC), providing a lower-bound of performance. An oracle agent (Oracle) that receives access to the user's true preference, providing an upper bound of performance, and a variation of PROSE that is conceptually equivalent to CIPHER, PROSE$_{\text{CE}}$, but uses PROSE's improved prompt templates. PROSE$_{\text{CE}}$

uses a single LLM generation, a single inference step, and uses no preference consistency verification.

5.1.6 Results and Discussion

We present our main PLUME results in Table 5.2. Results on PRELUDE can be found in Appendix D.3.3. To compare tasks on generation quality with metrics on different scales, we use a percentile score, where 0% corresponds to the no preference conditioning baseline (NPC) and 100% to the Oracle baseline. Percent improvements are reported as the difference in scores on this scale. Overall, $\text{PROSE}_{\text{Full}}$ outperforms PROSE_{CE} by 12%, and CIPHER by 33%.

RQ1. In our first question, we set out to verify whether generating iterative candidate trajectories is beneficial to inferring preferences. Comparing PROSE to its ablated versions on the action/generation quality metric (PPCM), shows that each component of the iterative refinement process improves performance. Comparing PROSE with no comparison generation — PROSE_{u} — to PROSE with a single LLM-generated comparison generation — $\text{PROSE}_{u,a}$ — we observe that providing the comparison generations is beneficial when inferring preferences (3.8% mean improvement). This result supports the algorithmic decisions in [109, 277]. Allowing for multiple refinement steps provides a further increase in performance (Table 5.2: $\text{PROSE}_{u,a}$ vs. $\text{PROSE}_{u,a,S>1}$, 7.8% mean improvement). This can be explained by the LLM having more chances to infer correct preferences. Lastly, when comparing $\text{PROSE}_{u,a,S>1}$ to $\text{PROSE}_{\text{Full}}$ we see another 3.2% improvement. This highlights the benefits of updating candidates after each inference step using the newly inferred preferences. In all, iterative refinement provides a mean improvement of 14.8%.

RQ2. We investigate the benefit of verifying preferences by comparing PROSE to PROSE_{NV}. Here, we see a modest but consistent of 1.5% and 1.7% for Pref. Sim. and PPCM respectively when using preference consistency verification.

RQ3. While on average across LLMs PROSE outperforms CIPHER and all PROSE ablations, ICL outperforms PROSE. However, Figure 5.2 shows that PROSE's performance scales better with the quality of the underlying LLM (e.g., `Qwen2.5-72B-Instruct` vs. `GPT-4o`) than all baselines except Oracle. Notably, when using `GPT-4o`, PROSE outperforms ICL (1.35 vs 1.32 task mean Appendix D.3.2). We further investigate the benefits and limitations of PROSE and the learning baselines by comparing the performance across preference sets (Figure 5.3), and find that ICL excels on sets with the strongest structural preferences (e.g., Chat Forum Posts which includes “write in the style of a tweet”). In contrast, PROSE excels on the preference sets requiring a more nuanced understanding of tone (e.g., Paper Review, which includes “be sharply critical”). From examining logs, we notice that the LLMs are less adept at inferring encompassing structural preferences and often try to capture these preferences using multiple relevant, but imperfect preferences (e.g., “use emojis for emphasis”, “use 1-2 specific hashtags”) As PROSE and ICL seem to have complementary strengths, we combine the two ($\text{PROSE}_{\text{Full+ICL}}$) for a gain of 7.8%, 8.9%, and 51.1% over PROSE, ICL, and CIPHER when using `GPT-4o` as the agent’s LLM.

Human Evaluation To further validate the effectiveness of PROSE, we ran human evaluation with 16 participants (3 are ML researchers; 9 women and 7 men; age in [19, 58]). Participants completed a within subjects AB test comparing PLUME+ICL generations to ICL generations and PLUME+ICL generations to CIPHER generations. Participants evaluated the final LLM generations (i.e. the generation after seeing all previous demonstrations) across all five seeds for two different preference sets for the email task and two different preference sets for the summarization task. This leads to a total of 20 survey items per method comparison. We used the responses to compute a win rate for PLUME+ICL compared to each of ICL and CIPHER. For PLUME+ICL versus ICL, we see an average win rate of 69.4%. For PLUME+ICL versus CIPHER, we see an average win rate of 91.8%. The human evaluation results are in line with our synthetic evaluation results and support the effectiveness of the synthetic evaluation.

Discussion. Our results demonstrate that using iterative refinement and consistency verification improves over CIPHER in terms of preference description quality, generation quality, and performance stability (i.e., performance increases with the number of demonstrations, see Figure D.3). Additionally the performance difference between CIPHER and PROSE_{CE} highlights the impact of our prompt-tuning efforts. In this regard, PLUME’s prompts (Appendix D.6) can serve as a valuable starting point for extensions to other tasks, however, task-specific adaptations should be made.

Our results suggest that consistency verification provides only a modest improvement to PROSE. Therefore, to better understand its impact, we examine the learning logs and find that consistency verification effectively prunes irrelevant preferences—e.g.,“be concise and direct”— and preferences that overfit to specific passages— e.g., “ include personal details about characters”. However, the pruned preferences typically have minimal impact on the performance metrics as they rarely contradict the true preferences. Moreover, the pruned preferences do not drastically alter the generations as the orthogonal preferences often match the LLM’s default behavior while the overfit preferences become irrelevant and ignored. As such, the current metrics have difficulty measuring the presence of these irrelevant preferences. Nevertheless, we believe it is valuable to prune the irrelevant preferences, as they reduce the number of tokens required.

We find PROSE is competitive with and complementary to ICL while providing several advantages: (1) preferences are easier to interact with than a dataset of in-context examples as a user can view and modify the inferred preferences, (2) at inference time, PROSE requires approximately $\frac{1}{10}$ of the prompt tokens, and (3) the inferred preference description can benefit a wider range of tasks (e.g. human-agent collaboration [219], sample efficient imitation/reinforcement learning, and generating personalized preference pairs for RLAIF [356]).

Lastly, while developing PROSE, we learned the importance of phrasing the preference description in the LLM’s “own words”. We initially sorted the preference components by length before aggregation, however, this led to an average performance drop of 11% across tasks relative to keeping the LLM’s order for the preference components. This finding is inline with other work that shows that LLMs are sensitive to the order of list items [284, 15]. We believe future work investigating the impact of the ordering may yield useful insights.

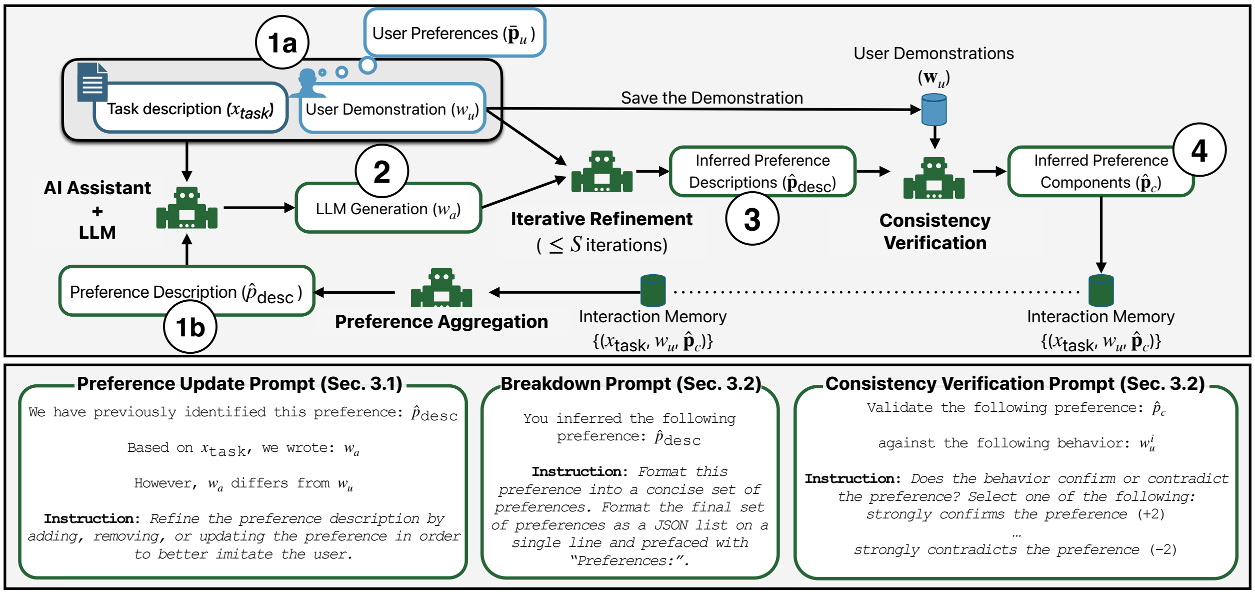
5.1.6.1 Limitations and Future Work

While PROSE and PLUME provide a number improvements, their limitations and challenges provide interesting avenues for future work. First, in this paper we focus on learning with the fewest user demonstrations possible. However another aspect of efficiency is the total number of tokens, and adding more refinement and preference consistency verification steps increases the number of tokens used. In our experiments, $\text{PROSE}_{\text{Full}}$ used 5.87x (prompt) and 6.07x (generated) more tokens on average than PROSE_{CE}. Given the monetary and environmental cost of LLMs, reducing the number of tokens while retaining performance is an important area for improvement. Lastly, a full-scale human trial would provide a greater understanding of the benefits and limitations of the proposed method. We look forward to investigating this more closely in future work.

5.1.7 Conclusion

In this work, we introduce PROSE, a novel algorithm for preference inference and refinement, along with a new benchmark, PLUME, designed to evaluate few-shot alignment of language models to individual users. PROSE draws inspiration from human hypothesis-driven learning: it iteratively refines a model's internal representation of user preferences by conditioning on updated preference descriptions and observing their effects, and it decomposes these preferences into interpretable components that are individually verified against demonstrations. This approach enables more accurate emulation of user behavior with minimal data, leading to improvements in alignment performance of up to 33

These results highlight how refinement mechanisms grounded in limited but informative observations can significantly enhance generalization in few-shot behavioral cloning. By modeling preference inference as a process of iterative hypothesis formation and testing, PROSE offers a path toward more personalized and adaptive AI systems that better reflect the human approach to learning from sparse data.


1a
User Preferences ($\bar{\mathbf{p}}_u$)
Task description (x_{task})
User Demonstration (w_u)
Save the Demonstration
User Demonstrations ($\mathbf{w}_u$)
AI Assistant + LLM
2
LLM Generation (w_a)
Inferred Preference Descriptions ($\hat{\mathbf{p}}_{desc}$)
Iterative Refinement ($\leq S$ iterations)
3
Consistency Verification
Inferred Preference Components ($\hat{\mathbf{p}}_c$)
4
Preference Description ($\hat{p}_{desc}$)
1b
Preference Aggregation
Interaction Memory $\{(x_{task}, w_u, \hat{\mathbf{p}}_c)\}$
Interaction Memory $\{(x_{task}, w_u, \hat{\mathbf{p}}_c)\}$

Preference Update Prompt (Sec. 3.1)

We have previously identified this preference: $\hat{p}_{desc}$

Based on x_{task}, we wrote: w_a

However, w_a differs from w_u

Instruction: *Refine the preference description by adding, removing, or updating the preference in order to better imitate the user.*

Breakdown Prompt (Sec. 3.2)

You inferred the following preference: $\hat{p}_{desc}$

Instruction: *Format this preference into a concise set of preferences. Format the final set of preferences as a JSON list on a single line and prefaced with "Preferences:".*

Consistency Verification Prompt (Sec. 3.2)

Validate the following preference: $\hat{p}_c$

against the following behavior: w_u^i

Instruction: *Does the behavior confirm or contradict the preference? Select one of the following:*
strongly confirms the preference (+2)
…
strongly contradicts the preference (-2)

Figure 5.1: **Overview of PROSE**. (top) The user provides a task description and demonstration to PROSE, which executes iterative refinement and then a consistency verification step. Iterative refinement updates the inferred preference description by generating a writing sample conditioned on the current preference description, comparing the sample to the user's demonstration, and updating the preference description to better describe the user's demonstration until the LLM's generations match the demonstration or a maximum number of iterations S is reached. The description is then broken into a set of component parts, and each component's consistency with prior demonstrations is verified with LLM-as-a-Judge. (bottom) Example PROSE prompts (for full prompts see Appendix D.6.1).

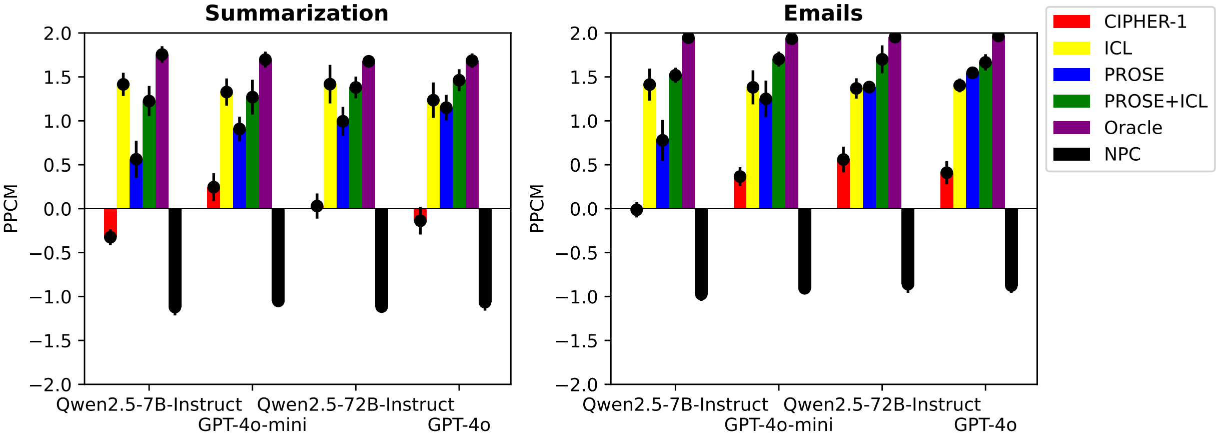
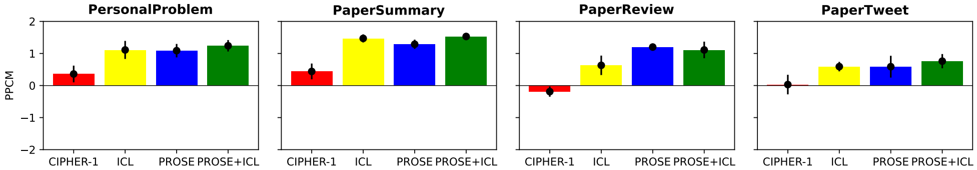
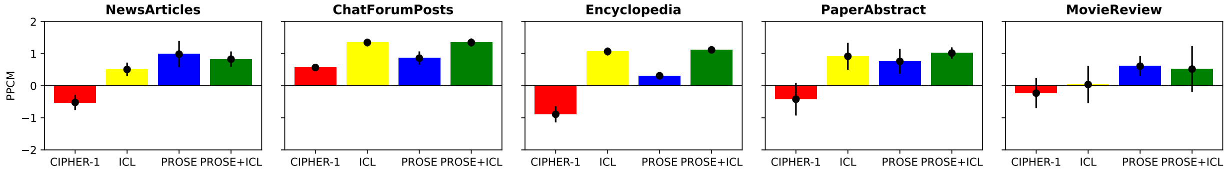


Figure 5.2: Preference compliance performance (PPCM) for CIPHER-1, in-context learning (ICL), PROSE, Oracle, and no preferences (NPC) for different preference-inferring LLMs. The LLMs are sorted by MMLU performance: `Qwen2.5-7B-Instruct` = 74.2, `GPT-4o-mini` = 82, `Qwen2.5-72B-Instruct` = 86.1, and `GPT-4o` = 88.7. `GPT-4o` is the proxy human with mean and standard error reported over 5 seeds.



(a) Email



(b) Summary

Figure 5.3: Generation quality (PPCM) for PROSE, CIPHER-1, in-context learning (ICL), and PROSE+ICL by **Email** (top) and **Summary** (bottom). `GPT-4o` is the agent's LLM with mean and standard error reported over 5 seeds.

Chapter 6

Discussion and Conclusion

6.1 Discussion

Throughout this thesis, I have explored strategies for enhancing AI generalization by taking inspiration from the current gold standard in adaptive intelligence: human cognition. I argue that generalization critically depends on three core elements: (1) the data used for learning, (2) the structure governing the flow of information, and (3) the mechanisms for updating the system. Humans excel at using their prior knowledge to identify salient streams of data, build sparse and causally linked structures to process this data, and leverage this world model to build and test hypotheses. By developing AI systems that emulate these core human strategies, we can achieve more robust and meaningful generalization.

In Chapter 3, I examine how the nature and composition of data shape AI generalization across two pivotal domains: natural language understanding and human-agent interaction. Inspired by the intrinsic relationship between language and embodied interaction in humans, I argue and empirically show that integrating multimodal data significantly enhances LLM capabilities. I first demonstrate the limitations of purely text-based training when reasoning about the physical world (Section 3.1). I then highlight reporting bias as a limiting factor in text-only training and illustrate how incorporating multimodal data mitigates these limitations (Section 3.2). Finally, I show how incorporating environmental context data in training substantially improves models' length generalization in sequential reasoning tasks (Section 3.3). Using similar insights on the importance of human data usage in the domain of human-age interaction, I illustrate how combining gameplay with eye gaze data enables faster and more accurate inference of critical teammate attributes such as proficiency, trust, and intent (Section 3.4).

Chapter 4 focuses on structural abstractions and their crucial role in generalization. Structures provide benefits through feature reuse and sparsity, which significantly reduces the search space. I highlight the particular importance of shared structures in collaborative settings. Inspired by how humans naturally employ structured task decompositions to create a common framework, I develop HA^2, a hierarchical reinforcement learning framework designed for human-agent collaboration. Results confirm that agents utilizing shared task abstractions generalize more effectively across varying environments and adapt more swiftly to both new human and artificial teammates (Section 4.1).

In Chapter 5, I explore hypothesis-driven learning mechanisms. Drawing inspiration from how humans build, test, and refine causal theories about the world, I introduce PROSE, a method designed to iteratively infer and refine latent user preferences from observed differences between expected and actual user outputs. PROSE significantly improves upon previous state-of-the-art methods by introducing iterative refinement and consistency verification, aligning system outputs more accurately and robustly with user preferences (Section 5.1).

6.2 Future Work

The insights presented in this thesis highlight promising avenues for developing more integrated, human-inspired AI systems. However, considerable work remains toward combining these strategies into cohesive frameworks.

Expanding on RESEED, future work could incorporate discrete latent representations to impose structural sparsity, as successfully demonstrated by methods such as DreamerV2 [136]. Furthermore, integrating explicit world models on top of LLM architectures could provide more structured predictions of sequential states that better resembles humans. Currently, RESEED tasks the LLM to both *encode* and *apply* sequential actions described in text. While the initial state is described in text, the remainder of the text, as is often the case, only describes actions. Introducing a world model that processes action representations generated by the LLM to predict subsequent states could further disentangle the problem and build even more on the concepts of hierarchy and abstraction [30]. Extending this approach to predict the necessary actions for achieving desired outcomes could further enable grounded, structured planning and type-2 reasoning capabilities [198]. Additionally, developing mechanisms to proactively identify and gather data addressing known uncertainties would offer another powerful strategy for improving system robustness and generalization.

With HA^2 we demonstrated how structure enables a strong foundation for generalizing to teammates. However, it still lacks a mechanism to learn new information about teammates. Combining HA^2 with our work on rapidly predicting salient teammate attributes from eye gaze and gameplay data is a clear next step toward targeted adaptations to new teammates within minimal interactions. Another interesting extension would be the inclusion of hypothesis-driven exploration to resolve ambiguities regarding teammate intentions, preferences, and capabilities. For example, if a teammate frequently engages in a specific task, targeted experimentation, such as temporarily shifting task responsibilities or directly querying the teammate, could clarify their motivations and preferences. This combined approach, drawing on both implicit observational data and explicit experimental interactions, aligns more closely with human adaptive strategies and could significantly enhance human-agent collaboration.

For PROSE, extending the hypothesis-driven learning framework beyond purely linguistic modalities to incorporate multimodal inputs, such as spatial reasoning or sensory-motor data, could greatly broaden its applicability and flexibility. Additionally, humans typically internalize explicit knowledge over time, converting it into habits and implicit knowledge to free cognitive resources. Similarly, employing methods such as Reinforcement Learning with AI Feedback (RLAIF) [21] could enable the gradual internalization of explicit insights directly into model parameters, thereby enhancing model efficiency and freeing computational resources to accommodate further learning and adaptation.

6.3 Conclusion

In this thesis, I demonstrate that human cognitive strategies—careful data selection, structured causal abstractions, and iterative hypothesis-driven learning—offer powerful templates for enhancing AI generalization. Across diverse domains including natural language understanding, sequential reasoning, human-agent collaboration, and behavioral cloning, I empirically validate the benefits of incorporating multimodal data, structured representations, and causal learning mechanisms into AI frameworks. These human-inspired approaches not only bolster generalization but also facilitate greater adaptability, interpretability, and alignment with human expectations and norms.

While the individual strategies presented demonstrate substantial improvements, the most significant potential lies in their integration. Future research should prioritize synthesizing sparse, hierarchical world models, targeted multimodal data strategies, and causal hypothesis-testing frameworks into unified systems. Ultimately, by closely modeling AI development on human cognitive principles, we pave the way toward creating artificial agents capable of the hallmark human intelligence: the flexible and robust generalization of knowledge across ever-changing contexts.

Bibliography

[1] Maria A. Rodriguez and Paola Merlo. Word associations and the distance properties of context-aware word embeddings. In Proceedings of the 24th Conference on Computational Natural Language Learning, pages 376–385, Online, 2020. Association for Computational Linguistics.

[2] Henny Admoni and Brian Scassellati. Social Eye Gaze in Human-Robot Interaction: A Review. Journal of Human-Robot Interaction, 6:25, 03 2017.

[3] Henny Admoni and Siddhartha S. Srinivasa. Predicting user intent through eye gaze for shared autonomy. In Proceedings of AAAI '16 Fall Symposium on Shared Autonomy in Research and Practice, pages 298 – 303, November 2016.

[4] Sanjeevan Ahilan and Peter Dayan. Feudal multi-agent hierarchies for cooperative reinforcement learning, 2019.

[5] Muneeb Ahmed, Brejesh Lall, Rajesh Kumar, and Arzad A. Kherani. Towards Estimation of Human Intent in Assistive Robotic Teleoperation Using Kinaesthetic and Visual Feedback. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, pages 1928–1934, October 2023.

[6] Stefano V. Albrecht and Subramanian Ramamoorthy. A game-theoretic model and best-response learning method for ad hoc coordination in multiagent systems. In Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, page 1155–1156, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems.

[7] Lorin W. Anderson and David R. Krathwohl, editors. A Taxonomy for Learning, Teaching, and Assessing. A Revision of Bloom's Taxonomy of Educational Objectives. Allyn & Bacon, New York, 2 edition, December 2001.

[8] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3674–3683, 2018.

[9] Felix Andlauer and Adam Laurell. Analysis of the delayed roll-out of fully autonomous vehicles. `https://www.drivesweden.net/en/project/analysis-delayed-roll-out-fully-autonomous-vehicles`, 2024. Drive Sweden Project Report.

[10] John Annett. Hierarchical task analysis. Handbook of cognitive task design, 2:17–35, 2003.

[11] Anonymous. Reseeding latent states for sequential language understanding. In Submitted to ACL Rolling Review - May 2025, 2025. under review.

[12] Anthropic. Introducing the next generation of claude, 2024.

[13] Stéphane Aroca-Ouellette, Miguel Aroca-Ouellette, Upasana Biswas, Katharina Kann, and Alessandro Roncone. Hierarchical reinforcement learning for ad hoc teaming. In Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '23, page 2337–2339, 2023.

[14] Stéphane Aroca-Ouellette, Natalie Mackraz, Barry-John Theobald, and Katherine Metcalf. Aligning LLMs by predicting preferences from user writing samples. In Forty-second International Conference on Machine Learning, 2025.

[15] Stéphane Aroca-Ouellette, Cory Paik, Alessandro Roncone, and Katharina Kann. PROST: Physical reasoning about objects through space and time. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4597–4608, Online, August 2021. Association for Computational Linguistics.

[16] Stéphane Aroca-Ouellette and Frank Rudzicz. On Losses for Modern Language Models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4970–4981, Online, November 2020. Association for Computational Linguistics.

[17] Stéphane Aroca-Ouellette, Miguel Aroca-Ouellette, Katharina von der Wense, and Alessandro Roncone. Implicitly aligning humans and autonomous agents through shared task abstractions, 2025.

[18] Reuben M. Aronson and Henny Admoni. Semantic gaze labeling for human-robot shared manipulation. In Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications, ETRA '19, New York, NY, USA, 2019. Association for Computing Machinery.

[19] Fabrizia Auletta, Rachel Kallen, Mario Di Bernardo, and Michael Richardson. Predicting and understanding human action decisions during skillful joint-action using supervised machine learning and explainable-AI. Scientific Reports, 13, 03 2023.

[20] Yu Awaya and Vijay Krishna. Communication and cooperation in repeated games. Theoretical Economics, 14(2):513–553, 2019.

[21] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. arXiv preprint arXiv:2212.08073, 2022.

[22] Anton Bakhtin, Laurens van der Maaten, Justin Johnson, Laura Gustafson, and Ross Girshick. Phyre: A new benchmark for physical reasoning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32, pages 5082–5093. Curran Associates, Inc., 2019.

[23] Nitsan Bar. Papertweet. `https://github.com/bnitsan/PaperTweet/`, 2022.

[24] Samuel Barrett, Avi Rosenfeld, Sarit Kraus, and Peter Stone. Making friends on the fly: Cooperating with new teammates. Artificial Intelligence, October 2016.

[25] Samuel Barrett, Peter Stone, and Sarit Kraus. Empirical evaluation of ad hoc teamwork in the pursuit domain. In Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS), May 2011.

[26] Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xlstm: Extended long short-term memory, 2024.

[27] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21, page 610–623, New York, NY, USA, 2021. Association for Computing Machinery.

[28] Emily M. Bender and Alexander Koller. Climbing towards NLU: On meaning, form, and understanding in the age of data. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5185–5198, Online, July 2020. Association for Computational Linguistics.

[29] Emily M. Bender and Alexander Koller. Climbing towards NLU: On meaning, form, and understanding in the age of data. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5185–5198, Online, July 2020. Association for Computational Linguistics.

[30] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell., 35(8):1798–1828, August 2013.

[31] Brent Berlin and Paul Kay. Basic Color Terms: Their Universality and Evolution. University of California Press, 1969.

[32] Chandra Bhagavatula, Jena D. Hwang, Doug Downey, Ronan Le Bras, Ximing Lu, Lianhui Qin, Keisuke Sakaguchi, Swabha Swayamdipta, Peter West, and Yejin Choi. I2D2: Inductive knowledge distillation with NeuroLogic and self-imitation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9614–9630, Toronto, Canada, July 2023. Association for Computational Linguistics.

[33] Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, et al. Experience grounds language. arXiv preprint arXiv:2004.10151, 2020.

[34] Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. Experience grounds language. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8718–8735, Online, November 2020. Association for Computational Linguistics.

[35] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020.

[36] B. S. Bloom, M. B. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl. Taxonomy of educational objectives. The classification of educational goals. Handbook 1: Cognitive domain. Longmans Green, New York, 1956.

[37] Paul Bloom. How children learn the meanings of words. MIT press, 2002.

[38] Valts Blukis, Dipendra Misra, Ross A. Knepper, and Yoav Artzi. Mapping navigation instructions to continuous control actions with position-visitation prediction. In Proceedings of The 2nd Conference on Robot Learning, page 505–518. PMLR, Oct 2018.

[39] Valts Blukis, Yannick Terme, Eyvind Niklasson, Ross A. Knepper, and Yoav Artzi. Learning to map natural language instructions to physical quadcopter control using simulated flight. In Proceedings of the Conference on Robot Learning, page 1415–1438. PMLR, May 2020.

[40] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Koulako Bala Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel J. Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. ArXiv, abs/2108.07258, 2021.

[41] George Boole. The mathematical analysis of logic. CreateSpace Independent Publishing Platform, 1847.

[42] Nick Bostrom. Ethical issues in advanced artificial intelligence. Machine Ethics and Robot Ethics, 2020.

[43] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT, pages 177–186. Springer, 2010.

[44] Jean-David Boucher, Ugo Pattacini, Amelie Lelong, Gerard Bailly, Frederic Elisei, Sascha Fagel, Peter Ford Dominey, and Jocelyne Ventre-Dominey. I reach faster when I see you look: gaze effects in human–human and human–robot face-to-face cooperation. Frontiers in neurorobotics, 6:3, 2012.

[45] Zied Bouraoui, José Camacho-Collados, and Steven Schockaert. Inducing relational knowledge from BERT. In The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 7456–7463. AAAI Press, 2020.

[46] Neil R. Bramley, Tobias Gerstenberg, Joshua B. Tenenbaum, and Todd M. Gureckis. Intuitive experimentation in the physical world. *Cognitive Psychology*, 105:9–38, 2018.

[47] Jake Brawer, Olivier Mangin, Alessandro Roncone, Sarah Widder, and Brian Scassellati. Situated human–robot collaboration: predicting intent from grounded natural language. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 827–833, 2018.

[48] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[49] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[50] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[51] Bernd Brügmann. Monte carlo go. Technical report, Technical report, Physics Department, Syracuse University Syracuse, NY, 1993.

[52] Kaylee Burns, Lisa Anne Hendricks, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. *ArXiv preprint*, abs/1803.09797, 2018.

[53] Eamonn Butler, Adam Smith, and Adam Smith. *The Condensed Wealth of Nations: And the incredibly condensed theory of moral sentiments*. Centre for Independent Studies, 2012.

[54] Rodrigo Canaan, Julian Togelius, Andy Nealen, and Stefan Menzel. Diverse agents for ad-hoc cooperation in hanabi. In *2019 IEEE Conference on Games (CoG)*, page 1–8. IEEE Press, 2019.

[55] Kate Candon, Jesse Chen, Yoony Kim, Zoe Hsu, Nathan Tsoi, and Marynel Vázquez. Nonverbal Human Signals Can Help Autonomous Agents Infer Human Preferences for Their Behavior. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, page 307–316, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems.

[56] Tianshi Cao, Jingkang Wang, Yining Zhang, and Sivabalan Manivasagam. Babyai++ : Towards grounded-language learning beyond memorization. In *ICLR*, 2020.

[57] Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-ai coordination. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[58] Thomas Carta, Clément Romac, Thomas Wolf, Sylvain Lamprier, Olivier Sigaud, and Pierre-Yves Oudeyer. Grounding large language models in interactive environments with online reinforcement learning. CoRR, abs/2302.02662, 2023.

[59] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, page 1721–1730, New York, NY, USA, 2015. Association for Computing Machinery.

[60] Loredana Caruccio, Stefano Cirillo, Giuseppe Polese, Giandomenico Solimando, Shanmugam Sundaramurthy, and Genoveffa Tortora. Claude 2.0 large language model: Tackling a real-world classification problem with a new iterative prompt engineering approach. Intelligent Systems with Applications, 21:200336, 2024.

[61] Tuhin Chakrabarty, Philippe Laban, Divyansh Agarwal, Smaranda Muresan, and Chien-Sheng Wu. Art or artifice? large language models and the false promise of creativity. In Proceedings of the CHI Conference on Human Factors in Computing Systems, pages 1–34, 2024.

[62] Rujikorn Charakorn, Poramate Manoonpong, and Nat Dilokthanakul. Investigating partner diversification methods in cooperative multi-agent deep reinforcement learning. In ICONIP, 2020.

[63] Feilong Chen, Duzhen Zhang, Minglun Han, Xiuyi Chen, Jing Shi, Shuang Xu, and Bo Xu. Vlp: A survey on vision-language pre-training. Machine Intelligence Research, 20:38–56, 2022.

[64] Shuo Chen, Ewa Andrejczuk, Zhiguang Cao, and Jie Zhang. Aateam: Achieving the ad hoc teamwork by employing the attention mechanism. Proceedings of the AAAI Conference on Artificial Intelligence, 34(05):7095–7102, Apr. 2020.

[65] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In European conference on computer vision, pages 104–120. Springer, 2020.

[66] Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. BabyAI: First steps towards grounded language learning with a human in the loop. In International Conference on Learning Representations, 2019.

[67] Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. In Advances in Neural Information Processing Systems 36, New Orleans, LA, USA, December 2023.

[68] B. Christian. The Alignment Problem: Machine Learning and Human Values. WW Norton, 2020.

[69] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[70] Herbert H. Clark and Susan E. Brennan. Grounding in communication. In Lauren Resnick, Levine B., M. John, Stephanie Teasley, and D., editors, Perspectives on Socially Shared Cognition, pages 13–1991. American Psychological Association, 1991.

[71] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. arXiv preprint arXiv:1803.05457, 2018.

[72] Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. On the use of arxiv as a dataset, 2019.

[73] Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao,

Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models, 2024.

[74] Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi ”Jim” Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick ”Tree” Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang,

Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. `https://arxiv.org/abs/2310.08864`, 2023.

[75] Russell Cooper, Douglas V. DeJong, Robert Forsythe, and Thomas W. Ross. Communication in the battle of the sexes game: Some experimental results. The RAND Journal of Economics, 20(4):568–587, 1989.

[76] Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. Textworld: A learning environment for text-based games. In Tristan Cazenave, Abdallah Saffidine, and Nathan Sturtevant, editors, Computer Games, pages 41–75, Cham, 2019. Springer International Publishing.

[77] Jacob W. Crandall, Mayada Oudah, Tennom, Fatimah Ishowo-Oloko, Sherief Abdallah, Jean-François Bonnefon, Manuel Cebrian, Azim Shariff, Michael A. Goodrich, and Iyad Rahwan. Cooperating with machines. Nature Communications, 9(1), jan 2018.

[78] Brandon Cui, Hengyuan Hu, Luis Pineda, and Jakob Foerster. K-level reasoning for zero-shot coordination in hanabi. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, Advances in Neural Information Processing Systems, volume 34, pages 8215–8228. Curran Associates, Inc., 2021.

[79] Jeff Da and Jungo Kasai. Cracking the contextual commonsense code: Understanding commonsense reasoning aptitude of deep contextual representations. In Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing, pages 1–12, Hong Kong, China, 2019. Association for Computational Linguistics.

[80] Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R. McKee, Joel Z. Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative ai, 2020.

[81] Gabrielle Davidson and Nicola Clayton. New perspectives in gaze sensitivity research. Learning & behavior, 44, 11 2015.

[82] Peter Dayan and Geoffrey E Hinton. Feudal reinforcement learning. In S. Hanson, J. Cowan, and C. Giles, editors, Advances in Neural Information Processing Systems, volume 5. Morgan-Kaufmann, 1992.

[83] Barry Devereux, Lorraine Tyler, Jeroen Geertzen, and Billi Randall. The centre for speech, language and the brain (cslb) concept property norms. Behavior research methods, 46, 2013.

[84] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[85] Bhavika Devnani, Skyler Seto, Zakaria Aldeneh, Alessandro Toso, Elena Menyaylenko, Barry-John Theobald, Jonathan Sheaffer, and Miguel Sarabia. Learning spatially-aware language and audio embedding, 2024.

[86] Thomas G. Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. J. Artif. Int. Res., 13(1):227–303, nov 2000.

[87] Edsger W Dijkstra. A note on two problems in connexion with graphs, 2022.

[88] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations, 2021.

[89] Jon Driver and Richard SJ Frackowiak. Neurobiological measures of human selective attention. Neuropsychologia, 39(12):1257–1262, 2001.

[90] Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding pretraining in reinforcement learning with large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, Proceedings of the 40th International Conference on Machine Learning, volume 202 of Proceedings of Machine Learning Research, pages 8657–8677. PMLR, 23–29 Jul 2023.

[91] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[92] E.H.S. Preface. In Edward Hance Shortliffe, editor, Computer-Based Medical Consultations: Mycin, pages xiii–xvi. Elsevier, 1976.

[93] Yanai Elazar, Abhijit Mahabal, Deepak Ramachandran, Tania Bedrax-Weiss, and Dan Roth. How large are lions? inducing distributions over quantitative attributes. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3973–3983, Florence, Italy, July 2019. Association for Computational Linguistics.

[94] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap learning audio concepts from natural language supervision. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

[95] Tore Ellingsen and Robert Östling. When does communication improve coordination? American Economic Review, 100(4):1695–1724, September 2010.

[96] Kutluhan Erol, James A Hendler, and Dana S Nau. Umcp: A sound and complete procedure for hierarchical task-network planning. In Aips, volume 94, pages 249–254, 1994.

[97] Allyson Ettinger. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. Transactions of the Association for Computational Linguistics, 8:34–48, 2020.

[98] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker. An integrated environment for the development of knowledge-based recommender applications. International Journal of Electronic Commerce, 12 2006.

[99] Richard E Fikes and Nils J Nilsson. Strips: A new approach to the application of theorem proving to problem solving. Artificial intelligence, 2(3-4):189–208, 1971.

[100] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, page 1126–1135. JMLR.org, 2017.

[101] Maxwell Forbes and Yejin Choi. Verb physics: Relative physical knowledge of actions and objects. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 266–276, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[102] Maxwell Forbes and Yejin Choi. Verb physics: Relative physical knowledge of actions and objects. In ACL, 2017.

[103] Wikimedia Foundation. Wikimedia downloads, 2022. Accessed: 2024-08-23.

[104] Maria Fox and Derek Long. Pddl2. 1: An extension to pddl for expressing temporal planning domains. Journal of artificial intelligence research, 20:61–124, 2003.

[105] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In ICLR. OpenReview.net, 2019.

[106] Chris D. Frith and Uta Frith. The neural basis of mentalizing. Neuron, 50(4):531–534, 2006.

[107] Iason Gabriel. Artificial intelligence, values, and alignment. Minds and Machines, 30(3):411–437, September 2020.

[108] Vittorio Gallese and George Lakoff. The brain's concepts: The role of the sensory-motor system in conceptual knowledge. Cognitive neuropsychology, 22(3-4):455–479, 2005.

[109] Ge Gao, Alexey Taymanov, Eduardo Salinas, Paul Mineiro, and Dipendra Misra. Aligning LLM agents by learning latent preference from user edits. In The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024.

[110] Ge Gao, Alexey Taymanov, Eduardo Salinas, Paul Mineiro, and Dipendra Misra. Aligning LLM agents by learning latent preference from user edits. In The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024.

[111] Divyansh Garg, Skanda Vaidyanath, Kuno Kim, Jiaming Song, and Stefano Ermon. LISA: Learning interpretable skill abstractions from language. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, Advances in Neural Information Processing Systems, 2022.

[112] Karl R Gegenfurtner and Jochem Rieger. Sensory and cognitive contributions of color to the recognition of natural scenes. Current Biology, 10(13):805–808, 2000.

[113] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In 2012 IEEE conference on computer vision and pattern recognition, pages 3354–3361. IEEE, 2012.

[114] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. Nature Machine Intelligence, 2(11):665–673, 2020.

[115] Susan A. Gelman. Learning from others: Children's construction of concepts. Annual Review of Psychology, 60:115–140, 2009.

[116] Dedre Gentner and Christian Hoyos. Analogy and abstraction. Topics in Cognitive Science, 9(3):672–693, 2017.

[117] M. Ghavamzadeh and S. Mahadevan. Learning to communicate and act using hierarchical reinforcement learning. In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004., pages 1114–1121, 2004.

[118] Mohammad Ghavamzadeh, Sridhar Mahadevan, and Rajbala Makar. Hierarchical multi-agent reinforcement learning. Autonomous Agents and Multi-Agent Systems, 13(2):197–229, sep 2006.

[119] Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In Proceedings of the NAACL Student Research Workshop, pages 8–15, San Diego, California, 2016. Association for Computational Linguistics.

[120] Arthur M Glenberg and Michael P Kaschak. Grounding language in action. Psychonomic bulletin & review, 9(3):558–565, 2002.

[121] Pranav Goel, Shi Feng, and Jordan Boyd-Graber. How pre-trained word representations capture commonsense physical comparisons. In Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing, pages 130–135, Hong Kong, China, November 2019. Association for Computational Linguistics.

[122] Gabriel Goh, Nick Cammarata †, Chelsea Voss †, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. Distill, 2021. https://distill.pub/2021/multimodal-neurons.

[123] Yoav Goldberg. Assessing bert's syntactic abilities. CoRR, abs/1901.05287, 2019.

[124] Nathalie Gontier. Depicting the tree of life: the philosophical and historical roots of evolutionary tree diagrams. Evolution Education and Outreach, 4:515–538, 09 2011.

[125] Alison Gopnik, Andrew Meltzoff, and Patricia Kuhl. The Scientist in the Crib: Minds, Brains, and How Children Learn. Harper Collins, 1999.

[126] Alison Gopnik and Andrew N Meltzoff. Words, thoughts, and theories. Mit Press, 1997.

[127] Alison Gopnik and Laura Schulz. Mechanisms of theory formation in young children. Trends in cognitive sciences, 8(8):371–377, 2004.

[128] Alison Gopnik and Henry M Wellman. Why the child's theory of mind really is a theory. Blackwell Publishing Ltd, 1992.

[129] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. IQA: Visual question answering in interactive environments. In Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on. IEEE, 2018.

[130] Jonathan Gordon and Benjamin Van Durme. Reporting bias and knowledge acquisition. In Proceedings of the 2013 workshop on Automated knowledge base construction, pages 25–30, 2013.

[131] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 6325–6334. IEEE Computer Society, 2017.

[132] Anna Grandori. Governance Structures, Coordination Mechanisms and Cognitive Models. Journal of Management & Governance, 1(1):29–47, March 1997.

[133] Simon Greipl, Katharina Bernecker, and Manuel Ninaus. Facial and Bodily Expressions of Emotional Engagement: How Dynamic Measures Reflect the Use of Game Elements and Subjective Experience of Emotions and Effort. Proc. ACM Hum.-Comput. Interact., 5(CHI PLAY), oct 2021.

[134] Thomas L Griffiths and Joshua B Tenenbaum. Theory-based causal induction. Psychological review, 116(4):661, 2009.

[135] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In International Conference on Learning Representations, 2020.

[136] Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In International Conference on Learning Representations, 2021.

[137] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. IEEE transactions on Systems Science and Cybernetics, 4(2):100–107, 1968.

[138] Jun Hatori, Yuta Kikuchi, Sosuke Kobayashi, Kuniyuki Takahashi, Yuta Tsuboi, Yuya Unno, Wilson Ko, and Jethro Tan. Interactively picking real-world objects with unconstrained spoken language instructions. In 2018 IEEE International Conference on Robotics and Automation (ICRA), page 3774–3781, May 2018.

[139] Susan J Hespos and Elizabeth S Spelke. Conceptual precursors to language. Nature, 430(6998):453–456, 2004.

[140] Felix Hill, Olivier Tieleman, Tamara von Glehn, Nathaniel Wong, Hamza Merzic, and Stephen Clark. Grounded language learning fast and slow, 2020.

[141] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. Neural computation, 18(7):1527–1554, 2006.

[142] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. Advances in neural information processing systems, 29, 2016.

[143] Guy Hoffman. Evaluating fluency in human–robot collaboration. IEEE Transactions on Human-Machine Systems, 49(3):209–218, 2019.

[144] Guy Hoffman. Evaluating Fluency in Human–Robot Collaboration. IEEE Transactions on Human-Machine Systems, 49(3):209–218, 2019.

[145] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, Advances in Neural Information Processing Systems, 2022.

[146] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. ACM Comput. Surv., 54(4), July 2021.

[147] Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. CoRR, abs/1709.01450, 2017.

[148] Chan-Jan Hsu, Hung-yi Lee, and Yu Tsao. XDBERT: Distilling visual information to BERT from cross-modal systems to improve language understanding. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 479–489, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[149] Joy Hsu, Jiayuan Mao, Joshua B. Tenenbaum, and Jiajun Wu. What's left? concept grounding with logic-enhanced foundation models. In Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[150] Catharine L. R. McGhan and Ali Nasir and Ella M. Atkins. Human intent prediction using markov decision processes. J. Aerosp. Inf. Syst., 12:393–397, 2012.

[151] Hengyuan Hu, Adam Lerer, Brandon Cui, Luis Pineda, Noam Brown, and Jakob Foerster. Off-belief learning. In Marina Meila and Tong Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 4369–4379. PMLR, 18–24 Jul 2021.

[152] Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "Other-play" for zero-shot coordination. In Hal Daumé III and Aarti Singh, editors, Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pages 4399–4410. PMLR, 13–18 Jul 2020.

[153] Xinyu Hua, Mitko Nikolov, Nikhil Badugu, and Lu Wang. Argument mining for understanding peer reviews. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2131–2137, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[154] Chien-Ming Huang, Sean Andrist, Allison Sauppé, and Bilge Mutlu. Using gaze patterns to predict task intent in collaboration. Frontiers in Psychology, 6, 2015.

[155] Shengyi Huang and Santiago Ontanon. A closer look at invalid action masking in policy gradient algorithms. The International FLAIRS Conference Proceedings, 35, 05 2022.

[156] Wenlong Huang, Fei Xia, Dhruv Shah, Danny Driess, Andy Zeng, Yao Lu, Pete Florence, Igor Mordatch, Sergey Levine, Karol Hausman, and brian ichter. Grounded decoding: Guiding text generation with grounded models for embodied agents. In Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[157] Yichen Huang and Lin F. Yang. Gemini 2.5 pro capable of winning gold at imo 2025, 2025.

[158] Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pages 6700–6709. Computer Vision Foundation / IEEE, 2019.

[159] Nikhil Hulle, Stéphane Aroca-Ouellette, Anthony J. Ries, Jake Brawer, Katharina Von Der Wense, and Alessandro Roncone. Eyes on the game: Deciphering implicit human signals to infer human proficiency, trust, and intent. In 2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN), pages 453–460, 2024.

[160] Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to train your robot with deep reinforcement learning: lessons we have learned. The International Journal of Robotics Research, 40(4-5):698–721, 2021.

[161] Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander T Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In 6th Annual Conference on Robot Learning, 2022.

[162] Max Jaderberg, Wojciech M. Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castañ eda, Charles Beattie, Neil C. Rabinowitz, Ari S. Morcos, Avraham Ruderman, Nicolas Sonnerat, Tim Green, Louise Deason, Joel Z. Leibo, David Silver, Demis Hassabis, Koray Kavukcuoglu, and Thore Graepel. Human-level performance in 3d multiplayer games with population-based reinforcement learning. Science, 364(6443):859–865, may 2019.

[163] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-z: Zero-shot task generalization with robotic imitation learning. In 5th Annual Conference on Robot Learning, 2021.

[164] Stanislaw Jastrzebski, Zac Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Amos Storkey, and Yoshua Bengio. Three factors influencing minima in SGD, 2018.

[165] Shervin Javdani, Henny Admoni, Stefania Pellegrinelli, Siddhartha S. Srinivasa, and J. Andrew Bagnell. Shared autonomy via hindsight optimization for teleoperation and teaming. The International Journal of Robotics Research, 37(7):717–742, 2018.

[166] Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, Fanzhi Zeng, Kwan Yee Ng, Juntao Dai, Xuehai Pan, Aidan O'Gara, Yingshan Lei, Hua Xu, Brian Tse, Jie Fu, Stephen Marcus McAleer, Yaodong Yang, Yizhou Wang, Song-Chun Zhu, Yike Guo, and Wen Gao. Ai alignment: A comprehensive survey. ArXiv, abs/2310.19852, 2023.

[167] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision, 2021.

[168] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know? Transactions of the Association for Computational Linguistics, 8:423–438, 2020.

[169] Woojeong Jin, Dong-Ho Lee, Chenguang Zhu, Jay Pujara, and Xiang Ren. Leveraging visual knowledge in language tasks: An empirical study on intermediate pre-training for cross-modal knowledge transfer. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2750–2762, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[170] Jun-Su Kang, Ukeob Park, Venkateswarlu Gonuguntla, Kalyana Chakravarthy Veluvolu, and Minho Lee. Human implicit intent recognition based on the phase synchrony of EEG signals. Pattern Recognition Letters, 66:144–152, 2015.

[171] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. arXiv preprint arXiv:2001.08361, 2020.

[172] Paul Kay, Brent Berlin, Luisa Maffi, William R Merrifield, and Richard Cook. The world color survey. CSLI Publications Stanford, CA, 2009.

[173] Harold H Kelley. Attribution theory in social psychology. In Nebraska symposium on motivation. University of Nebraska Press, 1967.

[174] Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[175] Daniel Kershaw and Rob Koeling. Elsevier OA cc-by corpus. CoRR, abs/2008.00774, 2020.

[176] Babatunde Keshinro, Younho Seong, and Sun Yi. Deep Learning-based human activity recognition using RGB images in Human-robot collaboration. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 66(1):1548–1553, 2022.

[177] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2022.

[178] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. UNIFIEDQA: Crossing format boundaries with a single QA system. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1896–1907, Online, November 2020. Association for Computational Linguistics.

[179] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In International conference on machine learning, pages 5583–5594. PMLR, 2021.

[180] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[181] Paul Knott, Micah Carroll, Sam Devlin, Kamil Ciosek, Katja Hofmann, Anca Dragan, and Rohin Shah. Evaluating the robustness of collaborative agents. In Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '21, page 1560–1562, Richland, SC, 2021. International Foundation for Autonomous Agents and Multiagent Systems.

[182] W Bradley Knox and Peter Stone. Tamer: Training an agent manually via evaluative reinforcement. In 2008 7th IEEE international conference on development and learning, pages 292–297. IEEE, 2008.

[183] Jongwoo Ko, Sungnyun Kim, Sungwoo Cho, and Se-Young Yun. Flex-judge: Think once, judge anywhere. arXiv preprint arXiv:2505.18601, 2025.

[184] Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. Transactions of the Association for Computational Linguistics, 6:317–328, 2018.

[185] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In European conference on machine learning, pages 282–293. Springer, 2006.

[186] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh, editors, Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pages 5338–5348. PMLR, 13–18 Jul 2020.

[187] Demetres Kostas, Stéphane Aroca-Ouellette, and Frank Rudzicz. Bendr: Using transformers and a contrastive self-supervised learning task to learn from massive amounts of eeg data. Frontiers in Human Neuroscience, 15, 2021.

[188] Christian Kothe, Seyed Yahya Shirazi, Tristan Stenner, David Medine, Chadwick Boulay, Matthew I. Grivich, Tim Mullen, Arnaud Delorme, and Scott Makeig. The lab streaming layer for synchronized multimodal recording. bioRxiv, 2024.

[189] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, Advances in Neural Information Processing Systems, volume 25. Curran Associates, Inc., 2012.

[190] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc., 2012.

[191] H. Kucera and W. N. Francis. Computational analysis of present-day American English. Brown University Press, Providence, RI, 1967.

[192] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, and et al. The open images dataset v4. International Journal of Computer Vision, 128(7):1956–1981, 2020.

[193] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[194] John E. Laird. The Soar Cognitive Architecture. MIT Press, 2008.

[195] Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. A Hierarchical Representation for Future Action Prediction. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, Computer Vision – ECCV 2014, pages 689–704, Cham, 2014. Springer International Publishing.

[196] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020.

[197] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[198] Yann LeCun. A path towards autonomous machine intelligence. OpenReview, 2022. Version 0.9.2, June 27, 2022.

[199] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[200] Youngwoon Lee, Jingyun Yang, and Joseph J. Lim. Learning to coordinate manipulation skills via skill behavior diversification. In International Conference on Learning Representations, 2020.

[201] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady, 10:707, February 1966.

[202] Kevin Leyton-Brown and Yoav Shoham. Essentials of Game Theory: A Concise, Multidisciplinary Introduction. Morgan and Claypool Publishers, 1st edition, 2008.

[203] Belinda Z. Li, Maxwell Nye, and Jacob Andreas. Language modeling with latent situations. In Findings of the Association for Computational Linguistics: ACL 2023, pages 12556–12571, Toronto, Canada, July 2023. Association for Computational Linguistics.

[204] Huao Li, Tianwei Ni, Siddharth Agrawal, Fan Jia, Suhas Raja, Yikang Gui, Dana Hughes, Michael Lewis, and Katia Sycara. Individualized mutual adaptation in human-agent teams. IEEE Transactions on Human-Machine Systems, 51(6):706–714, 2021.

[205] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, Advances in Neural Information Processing Systems, volume 36, pages 41451–41530. Curran Associates, Inc., 2023.

[206] Shengchao Li, Lin Zhang, and Xiumin Diao. Deep-Learning-Based Human Intention Prediction Using RGB Images and Optical Flow. Journal of Intelligent & Robotic Systems, 97:95 – 107, 2019.

[207] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16, pages 121–137. Springer, 2020.

[208] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In 2023 IEEE International Conference on Robotics and Automation (ICRA), pages 9493–9500. IEEE, 2023.

[209] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In 2023 IEEE International Conference on Robotics and Automation (ICRA), pages 9493–9500, 2023.

[210] Yancheng Liang, Daphne Chen, Abhishek Gupta, Simon Shaolei Du, and Natasha Jaques. Learning to cooperate with humans using generative agents. In The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024.

[211] Rensis Likert. A technique for the measurement of attitudes. Archives of psychology, 1932.

[212] Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. Birds have four legs?! NumerSense: Probing Numerical Commonsense Knowledge of Pre-Trained Language Models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6862–6868, Online, November 2020. Association for Computational Linguistics.

[213] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pages 740–755. Springer, 2014.

[214] Xiaoqiang Lin, Zhongxiang Dai, Arun Verma, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. Prompt optimization with human feedback, 2024.

[215] Yongjie Lin, Yi Chern Tan, and Robert Frank. Open sesame: Getting inside BERT's linguistic knowledge. In Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 241–253, Florence, Italy, 2019. Association for Computational Linguistics.

[216] Yuri Lin, Jean-Baptiste Michel, Erez Aiden Lieberman, Jon Orwant, Will Brockman, and Slav Petrov. Syntactic annotations for the Google Books NGram corpus. In Proceedings of the ACL 2012 System Demonstrations, pages 169–174, Jeju Island, Korea, 2012. Association for Computational Linguistics.

[217] Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. Transformer Circuits Thread, 2025.

[218] Tal Linzen. How can we accelerate progress towards human-like linguistic generalization? In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5210–5217, Online, July 2020. Association for Computational Linguistics.

[219] Jijia Liu, Chao Yu, Jiaxuan Gao, Yuqing Xie, Qingmin Liao, Yi Wu, and Yu Wang. Llm-powered hierarchical language agent for real-time human-ai coordination. In Mehdi Dastani, Jaime Simão Sichman, Natasha Alechina, and Virginia Dignum, editors, Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2024, Auckland, New Zealand, May 6-10, 2024, pages 1219–1228. International Foundation for Autonomous Agents and Multiagent Systems / ACM, 2024.

[220] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the Variance of the Adaptive Learning Rate and Beyond. In International Conference on Learning Representations, 2020.

[221] Ruibo Liu, Jason Wei, Shixiang Shane Gu, Te-Yen Wu, Soroush Vosoughi, Claire Cui, Denny Zhou, and Andrew M. Dai. Mind's eye: Grounded language model reasoning through simulation. In The Eleventh International Conference on Learning Representations, 2023.

[222] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692, 2019.

[223] Edward Lockhart, Neil Burch, Nolan Bard, Sebastian Borgeaud, Tom Eccles, Lucas Smaira, and Ray Smith. Human-agent cooperation in bridge bidding, 2020.

[224] Tania Lombrozo. Explanation and categorization: How "why?" informs "what?". Cognition, 110(2):248–253, 2009.

[225] Yi Loo, Chen Gong, and Malika Meghjani. A hierarchical approach to population training for human-ai collaboration. arXiv preprint arXiv:2305.16708, 2023.

[226] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In International Conference on Learning Representations, 2019.

[227] Xingzhou Lou, Jiaxian Guo, Junge Zhang, Jun Wang, Kaiqi Huang, and Yali Du. Pecan: Leveraging policy ensemble for context-aware zero-shot human-ai coordination. In Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '23, page 679–688, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems.

[228] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. Advances in neural information processing systems, 32, 2019.

[229] Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. NeuroLogic decoding: (un)supervised neural text generation with predicate logic constraints. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4288–4299, Online, June 2021. Association for Computational Linguistics.

[230] Yidu Lu and Nadine Sarter. Modeling and inferring human trust in automation based on real- time eye tracking data. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 64:344–348, 12 2020.

[231] Keane Lucas and Ross E. Allen. Any-play: An intrinsic augmentation for zero-shot coordination. In Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS '22, page 853–861, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems.

[232] Andrei Lupu, Brandon Cui, Hengyuan Hu, and Jakob Foerster. Trajectory diversity for zero-shot coordination. In Marina Meila and Tong Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 7204–7213. PMLR, 18–24 Jul 2021.

[233] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. Conference on Robot Learning (CoRL), 2019.

[234] Corey Lynch and Pierre Sermanet. Grounding language in play. arXiv preprint arXiv:2005.07648, 2020.

[235] Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. Robotics: Science and Systems, 2021.

[236] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[237] Rajbala Makar, Sridhar Mahadevan, and Mohammad Ghavamzadeh. Hierarchical multi-agent reinforcement learning. In Proceedings of the Fifth International Conference on Autonomous Agents, AGENTS '01, page 246–253, New York, NY, USA, 2001. Association for Computing Machinery.

[238] Olivier Mangin, Alessandro Roncone, and Brian Scassellati. How to be helpful? supportive behaviors and personalization for human-robot collaboration. Frontiers in Robotics and AI, 8:725780, 2022.

[239] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1906–1919, Online, July 2020. Association for Computational Linguistics.

[240] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. ArXiv e-prints, February 2018.

[241] Kevin R. McKee, Joel Z. Leibo, Charlie Beattie, and Richard Everett. Quantifying the effects of environment and population diversity in multi-agent reinforcement learning. Autonomous Agents and Multi-Agent Systems, 36(1), mar 2022.

[242] Donald McMillan, Barry A. T. Brown, Ikkaku Kawaguchi, Razan Jaber, Jordi Solsona Belenguer, and Hideaki Kuzuoka. Designing with Gaze: Tama - a Gaze Activated Smart-Speaker. Proc. ACM Hum. Comput. Interact., 3(CSCW):176:1–176:26, 2019.

[243] Oier Mees, Jessica Borja-Diaz, and Wolfram Burgard. Grounding language with visual affordances over unstructured data. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), London, UK, 2023.

[244] Oier Mees and Wolfram Burgard. Composing pick-and-place tasks by grounding language. In Bruno Siciliano, Cecilia Laschi, and Oussama Khatib, editors, Experimental Robotics, Springer Proceedings in Advanced Robotics, page 491–501, Cham, 2021. Springer International Publishing.

[245] Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data. IEEE Robotics and Automation Letters (RA-L), 7(4):11205–11212, 2022.

[246] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. IEEE Robotics and Automation Letters (RA-L), 7(3):7327–7334, 2022.

[247] Shakra Mehak, John D. Kelleher, Michael Guilfoyle, and Maria Chiara Leva. Action Recognition for Human–Robot Teaming: Exploring Mutual Performance Monitoring Possibilities. Machines, 12(1), 2024.

[248] Francisco Melo and Alberto Sardinha. Ad hoc teamwork by learning teammates' task. Autonomous Agents and Multi-Agent Systems, 30, 01 2015.

[249] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[250] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[251] John H Miller and Scott Moser. Communication and coordination. Complexity, 9(5):31–40, 2004.

[252] Henry Mintzberg. The Structuring of Organizations: A Synthesis of the Research. Prentice-Hall, 1979.

[253] Reuth Mirsky, Ignacio Carlucho, Arrasy Rahman, Elliot Fosong, William Macke, Mohan Sridharan, Peter Stone, and Stefano V. Albrecht. A survey of ad hoc teamwork research. In Dorothea Baumeister and Jörg Rothe, editors, Multi-Agent Systems, pages 275–293, Cham, 2022. Springer International Publishing.

[254] Dipendra Misra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi. Mapping instructions to actions in 3D environments with visual goal prediction. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2667–2678, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[255] Ishan Misra, C. Lawrence Zitnick, Margaret Mitchell, and Ross B. Girshick. Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 2930–2939. IEEE Computer Society, 2016.

[256] Gregory L. Murphy. The Big Book of Concepts. MIT Press, 2002.

[257] Stephen Neale. Paul grice and the philosophy of language. Linguistics and philosophy, pages 509–559, 1992.

[258] Douglas L Nelson, Cathy L McEvoy, and Thomas A Schreiber. The university of south florida free association, rhyme, and word fragment norms. Behavior Research Methods, Instruments, & Computers, 36(3):402–407, 2004.

[259] Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In Icml, volume 1, page 2, 2000.

[260] Truong-Huy Nguyen, David Hsu, Wee-Sun Lee, Tze-Yun Leong, Leslie Kaelbling, Tomas Lozano-Perez, and Andrew Grant. Capir: Collaborative action planning with intention recognition. In Proceedings of the 7th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2011, 10 2011.

[261] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. CoRR, abs/1803.02999, 2018.

[262] Stefanos Nikolaidis, Swaprava Nath, Ariel D. Procaccia, and Siddhartha Srinivasa. Game-theoretic modeling of human adaptation in human-robot collaboration. In Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction. ACM, mar 2017.

[263] Stefanos Nikolaidis and Julie A. Shah. Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy. 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 33–40, 2013.

[264] Stefanos Nikolaidis and Julie A. Shah. Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy. 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 33–40, 2013.

[265] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018.

[266] OpenAI. Faulty Reward Functions in The Wild, 2016. Accessed: 2024-11-10.

[267] OpenAI. Gpt-4 technical report. ArXiv, abs/2303.08774, 2023.

[268] OpenAI. Introducing OpenAI o1-preview, 2024. Accessed: 2024-11-10.

[269] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, Advances in Neural Information Processing Systems, 2022.

[270] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35:27730–27744, 2022.

[271] Cory Paik, Stéphane Aroca-Ouellette, Alessandro Roncone, and Katharina Kann. The World of an Octopus: How Reporting Bias Influences a Language Model‘s Perception of Color. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 823–835, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[272] Ronald Parr and Stuart Russell. Reinforcement learning with hierarchies of machines. In M. Jordan, M. Kearns, and S. Solla, editors, Advances in Neural Information Processing Systems, volume 10. MIT Press, 1997.

[273] Erika A. Patall, Nicole Yates, Jihyun Lee, Man Chen, Bethany H. Bhat, Kejin Lee, S. Natasha Beretvas, Shengjie Lin, Sophia Man Yang, Neil G. Jacobson, Eboneigh Harris, and Derek J. Hanson. A meta-analysis of teachers' provision of structure in the classroom and students' academic competence beliefs, engagement, and achievement. Educational Psychologist, 59(1):42–70, 2024.

[274] Shubham Pateria, Budhitama Subagdja, Ah-hwee Tan, and Chai Quek. Hierarchical reinforcement learning: A comprehensive survey. ACM Comput. Surv., 54(5), jun 2021.

[275] Judea Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, 1988.

[276] Judea Pearl. Causality. Cambridge University Press, 2 edition, 2009.

[277] Andi Peng, Andreea Bobu, Belinda Z Li, Theodore R Sumers, Ilia Sucholutsky, Nishanth Kumar, Thomas L Griffiths, and Julie A Shah. Preference-conditioned language-guided abstraction. In Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, pages 572–581, 2024.

[278] Andi Peng, Andreea Bobu, Belinda Z. Li, Theodore R. Sumers, Ilia Sucholutsky, Nishanth Kumar, Thomas L. Griffiths, and Julie A. Shah. Preference-conditioned language-guided abstraction. In Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, HRI '24, page 572–581, New York, NY, USA, 2024. Association for Computing Machinery.

[279] Andi Peng, Andreea Bobu, Belinda Z Li, Theodore R Sumers, Ilia Sucholutsky, Nishanth Kumar, Thomas L Griffiths, and Julie A Shah. Preference-conditioned language-guided abstraction. In Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, pages 572–581, 2024.

[280] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, 2014.

[281] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[282] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics.

[283] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference

on Natural Language Processing (EMNLP-IJCNLP), pages 2463–2473, Hong Kong, China, 2019. Association for Computational Linguistics.

[284] Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, Findings of the Association for Computational Linguistics: NAACL 2024, pages 2006–2017, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[285] Sandro Pezzelle. Dealing with semantic underspecification in multimodal NLP. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12098–12112, Toronto, Canada, July 2023. Association for Computational Linguistics.

[286] Jean Piaget. The Construction of Reality in the Child. Basic Books/Hachette Book Group, 1954. Translated from French by Margaret Cook.

[287] Patrick M. Pilarski, Andrew Butcher, Michael Johanson, Matthew M. Botvinick, Andrew Bolt, and Adam S. R. Parker. Learned human-agent decision-making, communication and joint action in a virtual reality environment, 2019.

[288] Dean A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D. Touretzky, editor, Advances in Neural Information Processing Systems, volume 1. Morgan-Kaufmann, 1988.

[289] Jan Pöppel, Sebastian Kahl, and Stefan Kopp. Resonating minds - emergent collaboration through hierarchical active inference. Cogn. Comput., 14(2):581–601, 2022.

[290] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8494–8502, 2018.

[291] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[292] Linlu Qiu, Hexiang Hu, Bowen Zhang, Peter Shaw, and Fei Sha. Systematic generalization on gSCAN: What is nearly solved and what is next? In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2180–2188, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[293] Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, S. M. Ali Eslami, and Matthew Botvinick. Machine Theory of Mind. In Jennifer Dy and Andreas Krause, editors, Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pages 4218–4227. PMLR, 10–15 Jul 2018.

[294] Alec Radford. Improving language understanding by generative pre-training, 2018.

[295] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 8748–8763. PMLR, 18–24 Jul 2021.

[296] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022.

[297] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019.

[298] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI, 2019. Accessed: 2024-11-15.

[299] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems, 36, 2024.

[300] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems, 36, 2024.

[301] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140):1–67, 2020.

[302] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. Journal of Machine Learning Research, 22(268):1–8, 2021.

[303] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[304] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[305] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125, 1(2):3, 2022.

[306] Yosef Razin and Karen Feigh. Learning to predict intent from gaze during robotic hand-eye coordination. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17, page 4596–4602. AAAI Press, 2017.

[307] J Redmon. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[308] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. Transactions of the Association for Computational Linguistics, 1:25–36, 2013.

[309] Marco Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In John DeNero, Mark Finlayson, and Sravana Reddy, editors, Proceedings of the

2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, pages 97–101, San Diego, California, June 2016. Association for Computational Linguistics.

[310] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4902–4912, Online, July 2020. Association for Computational Linguistics.

[311] Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. MCTest: A challenge dataset for the open-domain machine comprehension of text. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 193–203, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[312] Cintia Rodríguez, Pedro Palacios, Iván Moreno-Llanos, and Irene Guevara. The Social Life of Objects in Early Development. Springer, 2025.

[313] Alessandro Roncone, Olivier Mangin, and Brian Scassellati. Transparent role assignment and task allocation in human robot collaboration. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 1014–1021, 2017.

[314] Alessandro Roncone, Olivier Mangin, and Brian Scassellati. Transparent role assignment and task allocation in human robot collaboration. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 1014–1021. IEEE, 2017.

[315] Alessandro Roncone, Olivier Mangin, and Brian Scassellati. Transparent role assignment and task allocation in human robot collaboration. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 1014–1021, 2017.

[316] Alessandro Roncone, Ugo Pattacini, Giorgio Metta, and Lorenzo Natale. A Cartesian 6-DoF Gaze Controller for Humanoid Robots. In Robotics: science and systems, volume 2016, 2016.

[317] Isabelle Rosenthal, Sivalogeswaran Ratnasingam, Theodros Haile, Serena Eastman, Josh Fuller-Deets, and Bevil R. Conway. Color statistics of objects, and color tuning of object cortex in macaque monkey. Journal of Vision, 18(11):1–1, 2018.

[318] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Proceedings of the fourteenth international conference on artificial intelligence and statistics, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[319] S. Russell. Human Compatible: Artificial Intelligence and the Problem of Control. Penguin Publishing Group, 2019.

[320] Goutam Kumar Saha. Web ontology language (owl) and semantic web. Ubiquity, 2007(September), September 2007.

[321] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. Advances in neural information processing systems, 35:36479–36494, 2022.

[322] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large language models meet personalization, 2024.

[323] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In International Conference on Learning Representations, 2022.

[324] Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. Whose opinions do language models reflect? In International Conference on Machine Learning, pages 29971–30004. PMLR, 2023.

[325] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. CoRR, abs/1904.09728, 2019.

[326] Bidipta Sarkar, Andy Shih, and Dorsa Sadigh. Diverse conventions for human-AI collaboration. In Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[327] Albrecht Schmidt. Implicit Human Computer Interaction Through Context. Personal Technologies, 4, 07 1999.

[328] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. ArXiv, abs/1707.06347, 2017.

[329] Laura E Schulz and Elizabeth Baraff Bonawitz. Serious fun: preschoolers engage in more exploratory play when evidence is confounded. Developmental psychology, 43(4):1045, 2007.

[330] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In Regina Barzilay and Min-Yen Kan, editors, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[331] Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. The semantic web revisited. IEEE intelligent systems, 21(3):96–101, 2006.

[332] Omar Shaikh, Michelle Lam, Joey Hejna, Yijia Shao, Michael Bernstein, and Diyi Yang. Show, don't tell: Aligning language models with demonstrated feedback. arXiv preprint arXiv:2406.00888, 2024.

[333] Sumuk Shashidhar, Abhinav Chinta, Vaibhav Sahai, and Dilek Hakkani Tur. Unsupervised human preference learning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 3412–3445, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[334] Xiaoteng Shen, Rui Zhang, Xiaoyan Zhao, Jieming Zhu, and Xi Xiao. Pmg : Personalized multimodal generation with large language models, 2024.

[335] Yu Shi, Wei Xu, and Pingzhao Hu. Out of distribution learning in bioinformatics: advancements and challenges. Briefings in Bioinformatics, 26(3):bbaf294, 2025.

[336] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In Conference on robot learning, pages 894–906. PMLR, 2022.

[337] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, Proceedings of the 5th Conference on Robot Learning, volume 164 of Proceedings of Machine Learning Research, pages 894–906. PMLR, 08–11 Nov 2022.

[338] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[339] Vered Shwartz and Yejin Choi. Do neural language models overcome reporting bias? In Proceedings of the 28th International Conference on Computational Linguistics, pages 6863–6870, Barcelona, Spain (Online), 2020. International Committee on Computational Linguistics.

[340] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv preprint arXiv:1712.01815, 2017.

[341] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. Science, 362(6419):1140–1144, 2018.

[342] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. Nature, 550:354–, October 2017.

[343] Ronal Singh, Tim Miller, Joshua Newn, Liz Sonenberg, Eduardo Velloso, and Frank Vetere. Combining planning with gaze for online human intention recognition. In Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '18, page 488–496, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[344] Mark K. Singley and John R. Anderson. The transfer of cognitive skill. Harvard University Press, USA, 1989.

[345] N Slack, Alistair Brandon-Jones, and R Johnston. Operations Management, 8th edition. Pearson, 8th edition, June 2016.

[346] Steven Sloman and Steven A Sloman. Causal models: How people think about the world and its alternatives. Oxford University Press, 2009.

[347] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In Proceedings of the AAAI conference on artificial intelligence, volume 31, 2017.

[348] Neville A Stanton. Hierarchical task analysis: Developments, applications, and extensions. Applied ergonomics, 37(1):55–79, 2006.

[349] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. In Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[350] Peter Stone, Gal A. Kaminka, Sarit Kraus, and Jeffrey S. Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In Proceedings of the Twenty-Fourth Conference on Artificial Intelligence, July 2010.

[351] Peter Stone, Gal A. Kaminka, and Jeffrey S. Rosenschein. Leading a best-response teammate in an ad hoc team. In AAMAS Workshop on Agent Mediated Electronic Commerce, pages 153–167, May 2009.

[352] Peter Stone and Sarit Kraus. To teach or not to teach? decision making under uncertainty in ad hoc teams. In Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS, volume 1, pages 117–124, 01 2010.

[353] Shane Storks, Qiaozi Gao, Yichi Zhang, and Joyce Chai. Tiered reasoning for intuitive physics: Toward verifiable commonsense language understanding. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4902–4918, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[354] DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, Advances in Neural Information Processing Systems, volume 34, pages 14502–14515. Curran Associates, Inc., 2021.

[355] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE 2.0: A continual pre-training framework for language understanding. arXiv preprint arXiv:1907.12412, 2019.

[356] Zhiqing Sun, Yikang Shen, Hongxin Zhang, Qinhong Zhou, Zhenfang Chen, David Daniel Cox, Yiming Yang, and Chuang Gan. SALMON: Self-alignment with instructable reward models. In The Twelfth International Conference on Learning Representations, 2024.

[357] Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. In Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[358] Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. Artificial Intelligence, 112(1):181–211, 1999.

[359] Richard Stuart Sutton. Temporal credit assignment in reinforcement learning. University of Massachusetts Amherst, 1984.

[360] R.S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Artificial Intelligence, 112:181–211, 1999.

[361] Andrew Szot, Max Schwarzer, Harsh Agrawal, Bogdan Mazoure, Rin Metcalf, Walter Talbott, Natalie Mackraz, R Devon Hjelm, and Alexander T Toshev. Large language models as generalizable policies for embodied tasks. In The Twelfth International Conference on Learning Representations, 2024.

[362] Aaquib Tabrez, Matthew B. Luebbers, and Bradley Hayes. A Survey of Mental Modeling Techniques in Human–Robot Teaming. Current Robotics Reports, 2020.

[363] Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. olmpics-on what language model pre-training captures. Transactions of the Association for Computational Linguistics, 8:743–758, 2020.

[364] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. arXiv preprint arXiv:1908.07490, 2019.

[365] Hao Tan and Mohit Bansal. Vokenization: Improving language understanding with contextualized, visual-grounded supervision. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 2066–2080, Online, November 2020. Association for Computational Linguistics.

[366] Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. Democratizing large language models via personalized parameter-efficient fine-tuning, 2024.

[367] Tianyi Tang, Yushuo Chen, Yifan Du, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. Learning to imagine: Visually-augmented natural language generation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9468–9481, Toronto, Canada, July 2023. Association for Computational Linguistics.

[368] Zineng Tang, Jaemin Cho, Hao Tan, and Mohit Bansal. VidlanKD: Improving language understanding via video-distilled knowledge transfer. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, Advances in Neural Information Processing Systems, 2021.

[369] Shwetkranti Taware and Anuradha D. Thakare. Multimodal emotion recognition based on face and speech using deep convolution neural network and long short term memory. Circuits, Systems, and Signal Processing, 2025.

[370] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober,

Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, Iñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chain, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumaran, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz Kepa, Yomna Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezadegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakub Adamek, Tyler Mercado, Jonathan Mallinson, Siddhinita Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Älgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabinovitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejasi Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrc, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei "Louis" Chen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogev, Xiaochen Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen, Noah Ó Donnaile, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjot Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya

Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Ähdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji, Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzadkowski, Fiona Macintosh, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärrman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan, Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin Elsayed, Elnaz Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tumeh, Eyal Ben-David, Rishub Jain, Jonathan Uesato, Romina Datta, Oskar Bunyan, Shimu Wu, John Zhang, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu

Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshev, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei Li, Anoop Sinha, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Joe Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Buthpitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vít Listík, Mathias Carlen, Jan van de Kerkhof, Marcin Pikus, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirnschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhania, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh,

Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Tu Vu, Alek Andreev, Antoine He, Kevin Hui, Sheleem Kashem, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals. Gemini: A family of highly capable multimodal models, 2024.

[371] Lina Teichmann, Genevieve L. Quek, Amanda K. Robinson, Tijl Grootswagers, Thomas A. Carlson, and Anina N. Rich. The influence of object-color knowledge on emerging object representations in the brain. Journal of Neuroscience, 40(35):6779–6789, 2020.

[372] Joshua B Tenenbaum, Charles Kemp, Thomas L Griffiths, and Noah D Goodman. How to grow a mind: Statistics, structure, and abstraction. Science, 331(6022):1279–1285, 2011.

[373] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5238–5248, 2022.

[374] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011, pages 1521–1528. IEEE Computer Society, 2011.

[375] Patrizio E Tressoldi, Massimiliano Martinelli, Elisa Zaccaria, and Stefano Massaccesi. Implicit intuition: how heart rate can contribute to prediction of future events. Journal of the Society for Psychical research, 73(894):1, 2009.

[376] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2024.

[377] Paul Tylkin, Goran Radanovic, and David C. Parkes. Learning robust helpful behaviors in two-player cooperative atari environments. In Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '21, page 1686–1688, Richland, SC, 2021. International Foundation for Autonomous Agents and Multiagent Systems.

[378] Mike Uschold and Michael Gruninger. Ontologies: principles, methods and applications. The Knowledge Engineering Review, 11(2):93–136, 1996.

[379] Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models - a critical investigation. In Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[380] Emiel van Miltenburg. Stereotyping and bias in the flickr30k dataset. ArXiv preprint, abs/1605.06083, 2016.

[381] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

[382] Manuela M Veloso. Planning and learning by analogical reasoning. Springer, 1994.

[383] John Venn. Symbolic Logic. Macmillan, 1881.

[384] Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. FeUdal networks for hierarchical reinforcement learning. In Doina Precup and Yee Whye Teh, editors, Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 3540–3549. PMLR, 06–11 Aug 2017.

[385] Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Position: will we run out of data? limits of llm scaling based on human-generated data. In Proceedings of the 41st International Conference on Machine Learning, ICML'24. JMLR.org, 2024.

[386] Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 183–196, Online, 2020. Association for Computational Linguistics.

[387] Lennart Wachowiak, Peter Tisnikar, Gerard Canal, Andrew Coles, Matteo Leonetti, and Oya Celiktutan. Analysing Eye Gaze Patterns during Confusion and Errors in Human–Agent Collaborations. In RO-MAN 2022 - 31st IEEE International Conference on Robot and Human Interactive Communication, RO-MAN 2022 - 31st IEEE International Conference on Robot and Human Interactive Communication: Social, Asocial, and Antisocial Robots, pages 224–229, 2022.

[388] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. arXiv preprint 1905.00537, 2019.

[389] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. arXiv preprint 1804.07461, 2018.

[390] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. Transactions on Machine Learning Research, 2024.

[391] Lihui Wang, Robert Gao, József Váncza, Jörg Krüger, Xi Vincent Wang, Sotiris Makris, and George Chryssolouris. Symbiotic human-robot collaborative assembly. CIRP annals, 68(2):701–726, 2019.

[392] Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. {RODE}: Learning roles to decompose multi-agent tasks. In International Conference on Learning Representations, 2021.

[393] Weitian Wang, Rui Li, Yi Chen, and Yunyi Jia. Human Intention Prediction in Human-Robot Collaborative Tasks. In Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, HRI '18, page 279–280, New York, NY, USA, 2018. Association for Computing Machinery.

[394] Weizhi Wang, Li Dong, Hao Cheng, Haoyu Song, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Visually-augmented language modeling. In The Eleventh International Conference on Learning Representations, 2023.

[395] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. Transactions of the Association for Computational Linguistics, 9:176–194, 2021.

[396] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In The Eleventh International Conference on Learning Representations, 2023.

[397] Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. Blimp: The benchmark of linguistic minimal pairs for english. Transactions of the Association for Computational Linguistics, 8:377–392, 2020.

[398] Christopher JCH Watkins and Peter Dayan. Q-learning. Machine learning, 8:279–292, 1992.

[399] Nathaniel Weir, Adam Poliak, and Benjamin Van Durme. Probing neural language models for human tacit assumptions. In Proceedings of the Annual Meeting of the Cognitive Science Society, volume 42, 2020. Peer-reviewed. Retrieved from eScholarship.

[400] William F. Whitney, Min Jae Song, David Brandfonbrener, Jaan Altosaar, and Kyunghyun Cho. Evaluating representations by the complexity of learning low-loss predictors, 2021.

[401] Gwydion Williams. Hierarchical Influences on Human Decision-Making. PhD thesis, UCL (University College London), 2022.

[402] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing. ArXiv preprint, abs/1910.03771, 2019.

[403] Anpeng Wu, Kun Kuang, Minqin Zhu, Yingrong Wang, Yujia Zheng, Kairong Han, Baohong Li, Guangyi Chen, Fei Wu, and Kun Zhang. Causality for large language models. arXiv preprint arXiv:2410.15319, 2024.

[404] Feng Wu, Shlomo Zilberstein, and Xiaoping Chen. Online planning for ad hoc autonomous agent teams. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One, IJCAI'11, page 439–445. AAAI Press, 2011.

[405] Sarah A. Wu, Rose E. Wang, James A. Evans, Joshua B. Tenenbaum, David C. Parkes, and Max Kleiman-Weiner. Too many cooks: Coordinating multi-agent collaboration through inverse planning. Topics in Cognitive Science, n/a(n/a), 2021.

[406] Jiannan Xiang, Tianhua Tao, Yi Gu, Tianmin Shu, Zirui Wang, Zichao Yang, and Zhiting Hu. Language models meet world models: Embodied experiences enhance language models. In Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[407] Changnan Xiao and Bing Liu. Generalizing reasoning problems to longer lengths. In The Thirteenth International Conference on Learning Representations, 2025.

[408] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin

Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. arXiv preprint arXiv:2412.15115, 2024.

[409] Mingyu Yang, Yaodong Yang, Zhenbo Lu, Wengang Zhou, and Houqiang Li. Hierarchical multi-agent skill discovery. In Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[410] Yue Yang, Wenlin Yao, Hongming Zhang, Xiaoyang Wang, Dong Yu, and Jianshu Chen. Z-LaVI: Zero-shot language solver fueled by visual imagination. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 1186–1203, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[411] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. Advances in neural information processing systems, 32, 2019.

[412] JD Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, pages 1–21, 2023.

[413] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, Computer Vision – ECCV 2014, pages 818–833, Cham, 2014. Springer International Publishing.

[414] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 93–104, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[415] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics.

[416] Rowan Zellers, Ari Holtzman, Matthew Peters, Roozbeh Mottaghi, Aniruddha Kembhavi, Ali Farhadi, and Yejin Choi. PIGLeT: Language grounding through neuro-symbolic interaction in a 3D world. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 2040–2050, Online, August 2021. Association for Computational Linguistics.

[417] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, and Johnny Lee. Transporter networks: Rearranging the visual world for robotic manipulation. Conference on Robot Learning (CoRL), 2020.

[418] Kelly Zhang and Samuel Bowman. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 359–361, Brussels, Belgium, 2018. Association for Computational Linguistics.

[419] Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In International Conference on Learning Representations, 2020.

[420] Xinyuan Zhang, Mario Dalmaso, Luigi Castelli, Shimin Fu, and Giovanni Galfano. Cross-cultural asymmetries in oculomotor interference elicited by gaze distractors belonging to Asian and White faces. Scientific Reports, 11(1):20410, 2021.

[421] Xinyun Zhang, Haochen Tan, Han Wu, and Bei Yu. Towards versatile and efficient visual knowledge integration into pre-trained language models with cross-modal adapters, 2024.

[422] Yinsu Zhang, Aakash Yadav, Sarah K. Hopko, and Ranjana K Mehta. In Gaze We Trust: Comparing Eye Tracking, Self-report, and Physiological Indicators of Dynamic Trust during HRI. In Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, HRI '24, page 1188–1193, New York, NY, USA, 2024. Association for Computing Machinery.

[423] Yizhen Zhang, Minkyu Choi, Kuan Han, and Zhongming Liu. Explainable semantic space by grounding language to vision with cross-modal contrastive learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, Advances in Neural Information Processing Systems, 2021.

[424] Yizhen Zhang, Minkyu Choi, Kuan Han, and Zhongming Liu. Explainable semantic space by grounding language to vision with cross-modal contrastive learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, Advances in Neural Information Processing Systems, 2021.

[425] Michelle Zhao, Reid Simmons, and Henny Admoni. The role of adaptation in collective human–AI teaming. Topics in Cognitive Science, 2022.

[426] Rui Zhao, Jinming Song, Yufeng Yuan, Haifeng Hu, Yang Gao, Yi Wu, Zhongqian Sun, and Wei Yang. Maximum entropy population-based training for zero-shot human-ai coordination. Proceedings of the AAAI Conference on Artificial Intelligence, 37(5):6145–6153, Jun. 2023.

[427] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2023.

[428] Victor Zhong, Tim Rocktäschel, and Edward Grefenstette. Rtfm: Generalising to new environment dynamics via reading. In International Conference on Learning Representations, 2020.

[429] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. In Proceedings of the AAAI conference on artificial intelligence, volume 34, pages 13041–13049, 2020.

[430] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. arXiv preprint arXiv:2211.01910, 2022.

[431] Yue Maggie Zhou. Designing for complexity: Using divisions and hierarchy to manage complex tasks. Organization Science, 24(2):339–355, 2013.

[432] Y. Zhu, D. Gordon, E. Kolve, D. Fox, L. Fei-Fei, A. Gupta, R. Mottaghi, and A. Farhadi. Visual semantic planning using deep successor representations. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 483–492, 2017.

[433] Yuchen Zhuang, Haotian Sun, Yue Yu, Rushi Qiang, Qifan Wang, Chao Zhang, and Bo Dai. Hydra: Model factorization framework for black-box llm personalization, 2024.

[434] Maryam Ziaeefard and Freddy Lecue. Towards knowledge-augmented visual question answering. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, Proceedings of the 28th International Conference on Computational Linguistics, pages 1863–1873, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[435] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. arXiv preprint arXiv:1909.08593, 2019.

[436] Oleg Špakov, Howell Istance, Kari-Jouko Räihä, Tiia Viitanen, and Harri Siirtola. Eye gaze and head gaze in collaborative games. In the 11th ACM Symposium, pages 1–9, 06 2019.

Appendix A

Author Contributions for Presented Papers

A.1 PROST

Each author's contributions for *"PROST: Physical Reasoning about Objects through Space and Time"* [15] can be summarized as follows:

- Cory Paik contributed to code development, ran baseline experiments, generated result graphs and tables, and provided edits.
- Katharina von der Wense supported ideation, gave narrative feedback, and helped edit the manuscript.
- Alessandro Roncone contributed to paper editing.
- Stéphane Aroca-Ouellette conceived the core ideas, implemented the codebase, conducted experiments, developed the framing, created remaining figures, and authored the paper.

A.2 The World of an Octopus

Each author's contributions for *"The World of an Octopus: How Reporting Bias Influences a Language Model's Perception of Color"* [271] can be summarized as follows:

- Cory Paik co-led ideation, implemented the entire codebase, collected the data, ran baseline experiments, generated result graphs and tables, authored the related work and method sections, and edited all other sections.
- Katharina von der Wense co-led ideation, provided narrative guidance, and helped edit the writing.

- Alessandro Roncone contributed to paper editing.
- Stéphane Aroca-Ouellette co-led ideation, scoped and framed the paper—including the narrative and experimental design—authored the introduction, discussion, and conclusion, and edited other sections of the paper.

A.3 RESEED

Each author's contributions for *"ReSeeding Latent States for Sequential Language Understanding"* [11] can be summarized as follows:

- Katharina von der Wense supported ideation, provided narrative feedback, and contributed to editing.
- Alessandro Roncone supported ideation, provided narrative feedback, and contributed to editing.
- Stéphane Aroca-Ouellette conceived the core ideas, developed the codebase, ran experiments, created figures and tables, framed the paper, and authored the paper.

A.4 Eyes on the Game

Each author's contributions for *"Eyes on the Game: Deciphering Implicit Human Signals to Infer Human Proficiency, Trust, and Intent"* [159] can be summarized as follows:

- Nikhil Hulle developed the majority of the codebase, created tables and figures, and authored portions of the manuscript.
- Anthony J. Ries conducted all IRB-related tasks, co-designed and set up the data collection protocol, collected the data, and contributed to the writing.
- Katharina von der Wense served in a supervisory role.
- Alessandro Roncone contributed to ideation, provided narrative feedback, and helped edit the manuscript.

- Stéphane Aroca-Ouellette conceived the core ideas, co-designed the data collection protocol, guided code development and debugging, developed the framing of the paper, and authored key sections.

A.5 HA2

Each author's contributions for *"Implicitly Aligning Humans and Autonomous Agents through Shared Task Abstractions"* [17] can be summarized as follows:

- Katharina von der Wense provided narrative feedback.
- Miguel Aroca-Ouellette implemented the code to generate egocentric observations and produced Section 3.4.3.6 and Figure 4.4.
- Alessandro Roncone provided feedback and edits to improve the paper's framing.
- Stéphane Aroca-Ouellette conceived the core ideas, developed the codebase (including agent training and the user experiment platform), ran all experiments, established the initial framing, generated remaining figures, and authored the paper.

A.6 PROSE

Each author's contributions for *"Aligning LLMs by Predicting Preferences from User Writing Samples"* [14] can be summarized as follows:

- Rin Susa Metcalf played a major role in project scoping and design, implemented code to aggregate and visualize results, substantially reworked Figure 5.1, generated all appendix template documentation, and provided extensive edits to all sections of the paper.
- Natalie Mackraz contributed to project ideation, edited the paper for clarity and grammar, and provided moral support.
- Barry-John Theobald served in a supervisory role and supported project ideation.

- Stéphane Aroca-Ouellette conceived the core ideas, developed the codebase, conducted experiments, led the initial paper framing, generated other figures, drafted the manuscript, and collaboratively revised all sections with Rin.

Appendix B

Appendix for The World of an Octopus

B.1 Dataset Construction

B.2 Analysis of Annotator Biases

A potential side-effect of crowd-sourcing annotations is that annotators might be biased toward choosing fewer colors faster, as this would equate to higher monetary incentives. We observe a small correlation (Kendall's Tau=0.154, p=0.026) between the total time and number of colors selected. However, this is to be expected as selecting the colors takes time.

All models we evaluate were predominately trained on English text. To accommodate this domain and minimize dataset variance, we recruit only annotators from the United States. This may induce cultural or geographic biases: e.g., the color diversity of carrots is much smaller in the United States than in some Asian countries. Other geographic biases are more fine-grained; for example, the color of fire hydrants in the U.S. depends on where you live and the water source.

Additionally, our choice of colors is not as universal as, for example, the 6 color terms defined by The World Color Survey [172]. The latter may be more suitable for multilingual studies, though we leave such investigations for future work.

B.3 Experimental Details

For all experiments, we implement the CoDa dataset using the Huggingface Datasets Library. We use Huggingface's [402] pretrained models for evaluating all text-only models, and the official CLIP implementation by [295] for all CLIP models.[1] We run all experiments on a single machine with one Nvidia Titan RTX GPU.

B.3.1 Representation Probing

Our representation probing implementation is derived from the efficient JAX version provided by [400].[2]

We split the training set into 10 subsets spaced logarithmically from 1 to 311 objects, and report averages over 5 seeds. Note that for each seed, any additional points along the curve represent additional objects to the previous subset, however, different seeds have different object sets and thus a different number of samples per subset. For our dataset, we found the difference in samples to be far less impactful on performance than the number of objects.

All probes are 2-layer MLPs with ReLU activation functions and are trained using the Adam Optimizer [180] with a learning rate of 10^{-4}. All probes are trained for 4000 steps. More details on how to reproduce the experiments are provided in our GitHub repository.[3]

B.4 Zero Shot Results

The zero-shot results for all evaluated LMs are provided in Table B.1.

---

[1] github.com/openai/CLIP

[2] github.com/willwhitney/reprieve

[3] github.com/nala-cub/coda

Model		**Group**	**Spearman ρ ↑**	**Kendall's τ ↑**	**Acc@1 ↑**	**$\mathbf{D_{JS}}$ ↓**	**$\Delta\rho$ ↑**	**$\Delta\tau$ ↑**
GPT-2	B	Single	34.9 ± 25.7	28.8 ± 21.2	26.8	0.39 ± 0.07	−6.86	−6.60
		Multi	40.7 ± 22.9	32.9 ± 18.5	23.6	0.25 ± 0.06	−6.77	−5.46
		Any	**45.7 ± 25.0**	**36.9 ± 20.9**	**33.9**	**0.09 ± 0.04**	**2.17**	**2.64**
	M	Single	34.2 ± 26.4	28.5 ± 21.8	**35.9**	0.39 ± 0.07	−6.94	−6.30
		Multi	36.0 ± 26.0	29.5 ± 20.8	25.5	0.25 ± 0.06	−10.87	−8.51
		Any	**43.2 ± 26.5**	**34.8 ± 21.2**	35.7	**0.09 ± 0.04**	**0.09**	**0.74**
	L	Single	39.9 ± 24.4	32.8 ± 19.7	33.8	0.39 ± 0.07	−1.10	−1.91
		Multi	44.8 ± 20.9	36.5 ± 16.8	29.8	0.26 ± 0.06	−1.49	−1.05
		Any	**47.3 ± 26.9**	**37.9 ± 21.8**	**38.3**	**0.09 ± 0.04**	**4.35**	**3.95**
	XL	Single	40.3 ± 26.6	33.6 ± 22.1	**40.4**	0.39 ± 0.07	−0.55	−1.01
		Multi	41.7 ± 24.3	34.1 ± 19.4	28.8	0.25 ± 0.06	−4.66	−3.42
		Any	**48.1 ± 25.1**	**38.2 ± 20.2**	40.0	**0.09 ± 0.04**	**5.29**	**4.46**
RoBERTa	B	Single	41.5 ± 23.9	34.4 ± 19.6	**32.3**	0.32 ± 0.13	0.58	−0.21
		Multi	47.0 ± 21.9	37.7 ± 18.0	23.1	0.21 ± 0.09	0.44	−0.03
		Any	**51.9 ± 22.7**	**41.3 ± 18.9**	29.6	**0.11 ± 0.07**	**8.64**	**7.27**
	L	Single	47.8 ± 24.7	40.1 ± 20.8	**42.9**	0.28 ± 0.11	7.17	5.69
		Multi	50.2 ± 23.8	41.0 ± 19.5	33.2	0.19 ± 0.08	4.57	4.01
		Any	**52.5 ± 23.5**	**42.0 ± 19.5**	36.5	**0.10 ± 0.06**	**9.97**	**8.26**
ALBERT V1	B	Single	27.8 ± 25.0	23.2 ± 20.3	16.2	0.38 ± 0.10	−14.08	−12.30
		Multi	31.4 ± 24.2	25.1 ± 18.8	13.0	0.27 ± 0.09	−15.27	−12.60
		Any	**42.8 ± 26.4**	**33.7 ± 21.4**	**18.3**	**0.14 ± 0.06**	**-0.70**	**-0.63**
	L	Single	29.4 ± 27.0	24.3 ± 21.9	31.8	0.35 ± 0.13	−11.67	−10.62
		Multi	32.7 ± 22.6	26.7 ± 18.0	23.1	0.25 ± 0.09	−13.50	−10.78
		Any	**41.2 ± 25.7**	**33.6 ± 20.5**	**38.3**	**0.13 ± 0.06**	**-2.37**	**-0.58**
	XL	Single	36.4 ± 24.5	29.7 ± 20.0	26.3	0.35 ± 0.11	−4.73	−4.99
		Multi	44.6 ± 19.1	36.1 ± 15.5	26.9	0.22 ± 0.07	−1.53	−1.27
		Any	**48.2 ± 21.4**	**38.2 ± 17.2**	**35.7**	**0.11 ± 0.05**	**5.07**	**4.22**
	XXL	Single	39.9 ± 25.6	33.1 ± 21.1	31.3	0.31 ± 0.12	−1.38	−1.80
		Multi	41.3 ± 26.1	**33.2 ± 21.0**	23.6	0.21 ± 0.08	−5.23	−4.48
		Any	**41.9 ± 24.3**	32.8 ± 18.4	**38.3**	**0.11 ± 0.05**	**-0.87**	**-1.03**
ALBERT V2	B	Single	22.3 ± 29.7	18.9 ± 24.2	20.7	0.36 ± 0.11	−19.54	−16.46
		Multi	22.2 ± 26.8	18.0 ± 21.3	18.3	0.26 ± 0.07	−23.57	−19.11
		Any	**25.8 ± 26.9**	**20.8 ± 20.6**	**26.1**	**0.12 ± 0.05**	**-18.38**	**-13.98**
	L	Single	39.2 ± 27.1	32.5 ± 22.4	30.3	0.32 ± 0.11	**-2.14**	−2.50
		Multi	**41.9 ± 24.9**	**33.9 ± 20.4**	25.0	0.21 ± 0.07	−3.73	−3.10
		Any	40.0 ± 22.9	32.4 ± 18.1	**33.0**	**0.10 ± 0.05**	−3.70	**-2.05**
	XL	Single	25.2 ± 26.6	20.5 ± 21.5	**26.3**	0.35 ± 0.12	−16.14	−14.53
		Multi	25.4 ± 23.4	20.7 ± 18.4	23.6	0.25 ± 0.08	−20.51	−16.59
		Any	**29.6 ± 27.1**	**23.1 ± 20.8**	26.1	**0.12 ± 0.05**	**-12.48**	**-10.04**
	XXL	Single	43.7 ± 24.4	**36.4 ± 20.6**	34.3	0.30 ± 0.11	**2.69**	**1.55**
		Multi	**45.2 ± 23.7**	36.1 ± 19.5	25.5	0.20 ± 0.07	−1.08	−1.40
		Any	43.7 ± 24.9	34.5 ± 19.6	**39.1**	**0.10 ± 0.05**	0.95	0.70
Average		Single	35.9 ± 25.8	29.8 ± 21.2	30.7	0.35 ± 0.10	−5.33	−5.14
		Multi	38.9 ± 23.6	31.5 ± 19.0	24.5	0.24 ± 0.07	−7.37	−5.99
		Any	**43.0 ± 25.0**	**34.3 ± 19.9**	**33.5**	**0.11 ± 0.05**	**-0.14**	**0.28**

Table B.1: LM results when probed in a zero-shot setting. Single, Multi, and Any indicate sets of objects that are frequently a singe color, between two to four colors, or could be any color, respectively. All corr. coeff. (ρ, τ) are multiplied by 100. Means and std.s are calculated over objects of the respective group.

Appendix C

Appendix for RESEED

C.1 HOUSE details

Atomic Actions	**Low-Level Tasks**	**High-Level Tasks**
PickupObject	put_X_on_Y	stack_3
PutObject	put_X_in_Y	stack_4
OpenObject	heat_X	water_plants_using_X
CloseObject	fill_X	make_iced_coffee_in_X
ToggleObjectOn	brew_X	brew_tea_in_X
ToggleObjectOff	clean_X	toast_X
PourFromObject	slice_X	cook_X
SliceObject	pour_X_onin_Y	cook_and_remove_X
WipeObject	wipe_X_dry	clean_and_dry_X
	wipe_X_clean	clean_large_X

Table C.1: List of atomic actions, low-level tasks, and high-level tasks.

State Feature	**Related Affordance**
ObjectName (100)	–
isWet (2)	wettable
isCooked (2)	cookable
isClean (2)	cleanable
isFilledWithLiquid (2)	fillable
isOpen (2)	openable
isPickedUp (2)	pickupable
isSliced (2)	sliceable
isToggled (2)	toggleable
objectTemperature (3)	canChangeTemp
mass_change (3)	fillable (indirectly)
parentReceptaclesOn (6)	receptacleOn
parentReceptaclesIn (6)	receptacleIn

Table C.2: Mapping of state features to their related affordances. Number in (parentheses) denote the range of values the feature can take on.

HOUSE is a dataset inspired by the PigPen dataset used in [416]. We made the decision to adapt PigPen, rather than using the original dataset, for three primary reasons: 1) PigPen divides each full high-level task trajectory into a single (s_t, a_t, s_{t+1}) transition tuple, whereas we are interested in outcome of multiple sequential steps. 2) We found a range of inconsistencies and non-deterministic outcomes within the PigPen dataset (e.g. toast getting hot when turning ON the toaster in one instance, and the toast getting hot when turning OFF the toaster in one instance). 3) We wanted more control over the compositionality of tasks. A full list of actions and tasks is shown in Table C.1.

To this end, we manually crafted a deterministic transition function for each low-level action based on the affordances of each object and used it to create trajectories our trajectories. Matching ABCDs and CUBES, we use templates to create the language description of the trajectory. A full list of state features (used to encode the state) and affordances (unchangeable properties of objects which are not visible, but effect the outcome of the transition function) are shown in Table C.2.

C.2 ICL prompt

Figure C.1 outlines the full prompt used for in context learning. ¡¿ denote placeholder values. All examples came from the same distribution as the training set. For HOUSE, we ensured that a representative example for each of the 10 low-level tasks were used.

system >>>

You are tasked to solve sequential reasoning problems in which you will be given an initial state and a sequence of actions. Your job is to predict the final state after the sequence of actions is applied to the initial state. You will be given a list of examples that you can use to learn how to solve the problem. You must match the output format of the final state exactly. You will be graded on the exact accuracy of your predictions.

Expected output format:
<dataset specific formatting>.
Where terms enclosed in <> should be replaced with the actual output values.

user >>>

---Example 1---
Initial State and Actions:
<example 1 initial state & actions>

assistant >>>

Final State:
<example 1 final state>

...

user >>>

---Example 10---
Initial State and Actions:
<example 10 initial state & actions>

assistant >>>

Final State:
<example 10 final state>

C.3 Two-Step Training

To explore a more computationally friendly approach, we test a variation of RESEED, named that fully separates the alignment step from the generation step. Specifically, on alternating batches, we either perform a forward and backwards pass using the alignment losses, or we perform a forward pass with no gradients to generate Z' and then use those for the forward pass with gradients which decodes the final state description. A comparison of results between these two approaches is shown in Table C.3. While separating the alignment and generation steps does slightly reduce the performance of RESEED, it still outperforms the *TO*model, and does so with no additional memory requirements.

Model	ABCDs	CUBES	HOUSE
`Text-Only`	24.3 ± 0.1	37.9 ± 0.9	56.1 ± 5.0
RS$_{\texttt{Single Pass}}$	**100.0** ± 0.0	**65.0** ± 0.9	**75.7** ± 1.7
RS$_{\texttt{Separate Passes}}$	99.8 ± 0.2	64.6 ± 1.5	70.5 ± 2.9

Table C.3: Comparison of RESEED (RS) with and without separate backward passes. Results are the average accuracy and standard error across 5 seeds.

C.4 Additional Implementation Details

C.4.1 Hyper parameters

To tune the *TO*, we performed a grid search on the learning rate ($lr \in \{1e-5, 3e-5, 1e-4\}$, batch size ($bs \in \{32, 64, 128\}$), decay rate (per update step) ($dr \in \{0.9999, 0.99995, 0.99999\}$, and warm up steps ($ws \in \{400, 1000, 2000\}$). We found that across datasets, a batch size of 64 and 1000 warm up steps consistently provided the best results. For learning rate, we found $1e-4$ performed best on CUBES and HOUSE, while a learning rate of $3e-5$ performed best on ABCDs. For decary rate, we found 0.99995 performed best on CUBES and HOUSE, while a decay rate of 0.99999 performed best on ABCDs.

We used the same above hyper parameters for RESEED, only tuning $h_{dim} \in \{16, 32, 64, 128, 256, 512\}$ for each dataset. We found $h_{dim} = 16$, $h_{dim} = 128$, and $h_{dim} = 256$ performed best ABCDs, CUBES, and HOUSE respectively.

For the in-context learning LLMs used, all in-context learning examples came from the training set and were manually verified to be representative examples. The GPT-4o-mini checkpoint used was: gpt-4o-mini-2024-07-18, and the GPT-4o checkpoint used was gpt-4o-2024-08-06.

The five random seeds used were: [9590, 1282, 5742, 4674, 2921].

C.4.2 Computational Budget

All experiments were run using a single A100 (all experiments fit on a 40GB A100, although 80GB A100s were used as well). To reach convergence on a single run took between 10 minutes (1024 samples) and 16 hours (262144 samples). Compared to the *TO* model, RESEED took between 1.1x and 2x the amount of time to reach convergence. The additional cost is primarily due to the two forward passes, although on the ABCDs dataset, RESEED reached convergence much faster, mitigating the cost substantially. The experiments in this paper involved 20 runs per seed per dataset (with 5 seeds and 3 datasets), for a total of 300 runs.

C.5 AI assistant use

Claude-3.7-Sonnet-Thinking [12] was used to develop small portions of the code base. GPT-4o [267] was used as to edit the text at a paragraph level. All code and writing output from AI assistants were manually verified and edited as necessary by a human before use.

C.6 Additional Attributions and Attribution Info

The seed icon used in Figure 4.2 is from Flaticon.com. All artifacts were used in a manner consistent with their intended use.

Appendix D

Appendix for PROSE

D.1 Algorithm

Algorithm 1 Assistant Task Completion

```
Require: $x_{\text{task}}$ {Task instance}
Initialize empty preference set: $\hat{P}_c \leftarrow \varnothing$
Retrieve relevant examples:
    $E \leftarrow$ get_relevant_examples(interaction memory)
for each $e \in E$ do
    $\hat{P}_c \leftarrow \hat{P}_c \cup e.\hat{p}_c$
end for
Aggregate condense preferences:
    $\hat{p}_{desc} \leftarrow$ generate($\texttt{llm}, x_{\text{aggregate}}, \hat{P}_c$)
Sample agent generation:
    $w_a^0 \leftarrow$ generate($\texttt{llm}, x_{\text{task}}, \hat{p}_{desc}$)
Return Completed generation $w_a^0$, preference description $\hat{p}_{desc}$, and relevant examples $E$
```

Algorithm 2 PROSE: Preference Reasoning by Observing and Synthesizing Examples

```
Require: $x_{\text{task}}$ {Task instance}
Require: $w_u$ {User demonstration}
Require: $w_a^0$ {Agent generation}
Require: $\hat{p}_{desc}$ {Preference description}
Require: $E$ {relevant Examples}
Initialize $\hat{P}_c \leftarrow \texttt{generate}(\texttt{llm}, x_{\text{breakdown}}, \hat{p}^0_{\text{desc}})$
for each $s \in [0, S]$ do
    if $w_a^s = w_u$ then
        Stop refinement
    else
        Refine preferences:
            $\hat{p}^{s+1}_{\text{desc}} = \texttt{generate}(\texttt{llm}, x_{\text{update}}, \hat{p}^s_{\text{desc}}, w_u, w_a^s)$
        Decompose preference:
            $\hat{P}_c \leftarrow \hat{P}_c \cup \texttt{generate}(\texttt{llm}, x_{\text{breakdown}}, \hat{p}^s_{\text{desc}})$
        Generate new candidate generation:
            $w_a^s \leftarrow \texttt{generate}(\texttt{llm}, x_{\text{task}}, \hat{p}_{desc})$
    end if
end for
Initialize empty verification score list:
    $v_{scores} \leftarrow \varnothing$
for each $\hat{p}_c$ in $\hat{P}_c$ do
    for each $e \in E$ do
        Verify preference against demonstration:
            $v_{scores} \leftarrow v_{scores} \cup \texttt{generate}(\texttt{llm}, x_{\text{verification}}, e.w_u^i, \hat{p}^s_{\text{desc}})$
    end for
    if $\texttt{mean}(v_{scores}) < v$ then
        Discard $\hat{p}_c$ from $\hat{P}_c$
    end if
end for
Add $(x_{\text{task}}, w_u^i, \hat{P}_c)$ to interaction memory
```

D.2 Metric Definitions

D.2.1 Preference Inference Quality

Preference Description Length As conditioning on unnecessary tokens when generating responses aligned with user preferences is undesirable, we measure the number of tokens in the preference description. The preference length (Pref Len) is the number of characters used to describe a user's preferences, which is highly correlated with the number of tokens required.

Preference Similarity To assess the similarity between the inferred preferences and the ground truth preferences, the human proxy (`GPT-4o` in this paper) is prompted to evaluate how similar each inferred preference is to each ground truth preference following:

$$\text{Preference Similarity} = \text{llm\_judge}(\text{true}, \text{inferred}), \tag{D.1}$$

where true is the true preferences (see Appendix D.4 Table D.7), inferred is the inferred preference description, and llm_judge is a function that prompts the human proxy LLM to evaluate how well a given inferred preference aligns with the true preference on a scale of 0 to +4 (see Appendix Figure D.1).

D.2.2 Generation Quality

To assess the quality of the preference conditioned LLM's generations, the human proxy (`GPT-4o` in this paper) is prompted to evaluate how well the given generation complies with the each of the ground truth user preferences. The generation quality is then compute as the mean score over ground truth user preference components following:

$$\text{PPCM} = \frac{\sum_i^{|\text{true}|} \text{llm\_judge}(\text{true}_i, \text{assistant\_attempt})}{|\text{true}|}, \tag{D.2}$$

where true is the set of true preferences (see Appendix D.4 Table D.7), assistant_attempt is the assistant's summary or email, and llm_judge is a function that prompts the human proxy LLM to evaluate how well a given assistant solution aligns with the true preference on a scale of -2 to +2 (see Figure D.2).

The prompt used by the LLM-as-a-Judge for generation quality evaluation are shown in Appendix Figure D.2.

System Prompt

You are an experienced editor that is evaluating how similar writing preferences are.

User Prompt

You received the following description of a user’s writing preferences:
“““
Inferred preference: `<inferred_preference_i>`
”””
”How similar are the inferred preferences to the true writing preferences below?
True preference: `<true_preference_i>`?
Analyze how the preferences would impact a user’s writing. After reasoning, select one of the following options:
extremely similar, very similar, moderately similar, slightly similar, not at all similar
Your final selection should be on a new line prefaced with “Verdict:”

Figure D.1: **LLM-as-a-Judge prompts** to assess the similarity between the true and inferred preferences. The system prompt is prepended to the user prompt following the LLM’s chat template. “`<...>`” indicates that the text is formatted from a variable. `inferred_preference_i` one of the inferred preferences. `true_preference_i` refers to one of the k true preferences that the user has.

System Prompt

You are an experienced editor that is evaluating writing samples.

User Prompt

You received the following {`summary` | `email`}:
“““
`<agent_completion>`
”””
Does the above {`summary` | `email`} exhibit the following preference: `<true_preference_i>`?
Identify, analyze, and reason about specific excerpts that show similarities or contradictions of underlying preferences. After reasoning, select one of the following options:
clearly exhibits, somewhat exhibits, neither exhibits nor contradicts, somewhat contradicts, clearly contradicts
Your final selection should be on a new line prefaced with “Verdict:”

Figure D.2: **LLM-as-a-Judge prompts** for the per preference-component match metric (PPCM) used in the **PLUME environment**. The system prompt is prepended to the user prompt following the LLM’s chat template. “`<...>`” indicates that the text is formatted from a variable. `agent_completion` refers to the agent’s article summary or email, depending on the sub-task. `true_preference_i` refers to one of the k true preferences that the user has.

D.3 Extended Results

Additional results tables and figures discussed in the main body of the paper.

D.3.1 Metric Correlation

The metric correlation results for the assistive writing tasks (Table D.1).

	PRELUDE				PRELUDE $_{NoEdit}$				PLUME			
Metric	Acc.	B.Score	P. Len.	P. Sim.	Acc.	B.Score	P. Len.	P. Sim.	Acc.	B.Score	P. Len.	P. Sim
Summarization												
L-dist	-0.47	-0.53	-0.43	-0.46	-0.08	-0.14	0.01	-0.18	0.02	-0.13	-0.21	-0.10
ln-L-dist	-0.50	-0.57	-0.48	-0.51	-0.22	-0.36	-0.23	-0.42	-0.08	-0.33	-0.40	-0.36
PPCM	0.49	**0.63**	0.53	0.54	0.51	**0.65**	0.54	0.58	0.34	0.65	**0.71**	**0.71**
Emails												
L-dist	-0.26	-0.32	-0.27	-0.26	-0.10	-0.31	-0.36	-0.27	-0.11	-0.19	-0.25	-0.19
ln-L-dist	-0.23	-0.31	-0.27	-0.26	-0.08	-0.34	-0.42	-0.28	-0.12	-0.34	-0.41	-0.38
PPCM	0.18	0.30	**0.43**	0.18	0.20	0.32	**0.45**	0.19	0.48	**0.79**	0.74	**0.79**
Across Both Tasks												
L-dist	-0.43	-0.43	-0.31	-0.39	-0.09	-0.17	-0.03	-0.20	0.01	-0.15	-0.21	-0.11
ln-L-dist	-0.45	-0.45	-0.32	-0.42	-0.18	-0.27	-0.08	-0.32	-0.09	-0.32	-0.39	-0.35
PPCM	0.42	**0.48**	0.37	0.42	0.45	0.52	0.39	**0.46**	0.39	0.68	0.71	**0.73**

Table D.1: Pearson R correlation between preference similarity metrics and generated writing similarity metrics broken down by task (summarization vs email). For Levenshtein distance (L-dist) and length-normalized Levenshtein distance (ln-L-dist) lower is better, so inverse correlation is expected. For PPCM, Acc. (Accuracy), B.SCore (BertScore), P. Len (Preference Description Length), and P. Sim. (Preference Description Similarity) higher is better. Best correlation in each framework is bold. Best overall correlation is underlined. See Appendix D.2 for a full description of each metric.

D.3.2 Baselines and Ablations per LLM

The baseline and PROSE ablation results for `Qwen2.5-7b-Instruct`, `Qwen2.5-72b-Instruct`, `GPT-4o-mini`, and `GPT-4o`. The results in Table 5.2 are averaged across these four LLMs and results reported in this section.

	Summarization			Emails		
Method	Pref Len	Pref. Sim.	PPCM	Pref Len	Pref. Sim.	PPCM
No Learning Baselines						
NPC	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$-1.12_{\pm 0.04}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$-0.97_{\pm 0.04}$
Oracle	$120.80_{\pm 0.00}$	$3.88_{\pm 0.05}$	$1.75_{\pm 0.04}$	$118.50_{\pm 0.00}$	$3.90_{\pm 0.06}$	$1.95_{\pm 0.01}$
Learning Baselines						
ICL	$6237.54_{\pm 1163.61}$	$0.00_{\pm 0.00}$	$\mathbf{1.42}_{\pm \mathbf{0.06}}$	$6754.73_{\pm 975.36}$	$0.00_{\pm 0.00}$	$1.41_{\pm 0.08}$
CIPHER-1	$50.40_{\pm 1.68}$	$0.90_{\pm 0.04}$	$-0.33_{\pm 0.04}$	$51.55_{\pm 0.93}$	$1.32_{\pm 0.10}$	$-0.01_{\pm 0.04}$
CIPHER-5	$\mathbf{49.78}_{\pm \mathbf{1.86}}$	$\mathbf{1.04}_{\pm \mathbf{0.05}}$	$-0.11_{\pm 0.07}$	$\mathbf{49.04}_{\pm \mathbf{2.46}}$	$\mathbf{1.66}_{\pm \mathbf{0.05}}$	$-0.01_{\pm 0.06}$
PROSE Ablations						
PROSE$_{\text{CE}}$	$306.05_{\pm 15.72}$	$0.59_{\pm 0.03}$	$0.07_{\pm 0.06}$	$390.29_{\pm 20.64}$	$0.99_{\pm 0.05}$	$0.62_{\pm 0.06}$
PROSE$_{u}$	$274.05_{\pm 20.33}$	$0.45_{\pm 0.08}$	$-0.02_{\pm 0.05}$	$353.77_{\pm 33.87}$	$0.84_{\pm 0.05}$	$0.16_{\pm 0.15}$
PROSE$_{u,a}$	$291.45_{\pm 15.41}$	$0.61_{\pm 0.11}$	$0.03_{\pm 0.15}$	$301.27_{\pm 26.19}$	$1.00_{\pm 0.09}$	$0.45_{\pm 0.04}$
PROSE$_{u,a,S>1}$	$574.73_{\pm 27.75}$	$0.77_{\pm 0.06}$	$0.40_{\pm 0.11}$	$611.92_{\pm 32.87}$	$1.14_{\pm 0.06}$	$0.83_{\pm 0.05}$
PROSE$_{\text{NV}}$	$745.49_{\pm 30.03}$	$0.90_{\pm 0.07}$	$0.48_{\pm 0.10}$	$753.11_{\pm 75.90}$	$0.97_{\pm 0.11}$	$0.72_{\pm 0.04}$
PROSE$_{\text{Full}}$	$604.69_{\pm 31.19}$	$0.89_{\pm 0.10}$	$0.56_{\pm 0.09}$	$698.34_{\pm 44.28}$	$0.91_{\pm 0.09}$	$0.78_{\pm 0.10}$
PROSE$_{\text{Full+ICL}}$	$6829.38_{\pm 1159.22}$	$0.76_{\pm 0.07}$	$1.23_{\pm 0.08}$	$7435.39_{\pm 980.26}$	$0.96_{\pm 0.06}$	$\mathbf{1.52}_{\pm \mathbf{0.04}}$

Table D.2: `Qwen2.5-7b-Instruct` + PROSE's performance on the two tasks measured by the quality of inferred preferences (Pref. Sim.) and preference compliance (PPCM) compared against no preference generation (NPC), true preference generation (Oracle), in-context learning (ICL), CIPHER [109], and ablations over PROSE's components. Results are the mean and standard error across five seeds. Best non-Oracle results per task are bolded.

	Summarization			Emails		
Method	Pref Len	Pref. Sim.	PPCM	Pref Len	Pref. Sim.	PPCM
No Learning Baselines						
NPC	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$-1.11_{\pm 0.02}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$-0.86_{\pm 0.04}$
Oracle	$120.80_{\pm 0.00}$	$3.88_{\pm 0.08}$	$1.68_{\pm 0.03}$	$118.50_{\pm 0.00}$	$3.85_{\pm 0.06}$	$1.95_{\pm 0.01}$
Learning Baselines						
ICL	$6237.54_{\pm 1163.61}$	$0.00_{\pm 0.00}$	$\mathbf{1.42}_{\pm \mathbf{0.10}}$	$6754.73_{\pm 975.36}$	$0.00_{\pm 0.00}$	$1.37_{\pm 0.05}$
CIPHER-1	$\mathbf{88.36}_{\pm \mathbf{1.91}}$	$1.26_{\pm 0.04}$	$0.03_{\pm 0.06}$	$\mathbf{83.01}_{\pm \mathbf{2.03}}$	$\mathbf{1.82}_{\pm \mathbf{0.07}}$	$0.56_{\pm 0.07}$
CIPHER-5	$141.23_{\pm 6.64}$	$1.22_{\pm 0.05}$	$-0.06_{\pm 0.06}$	$141.45_{\pm 8.35}$	$1.60_{\pm 0.08}$	$0.34_{\pm 0.05}$
PROSE Ablations						
PROSE$_{\text{CE}}$	$359.01_{\pm 23.18}$	$1.25_{\pm 0.09}$	$0.62_{\pm 0.10}$	$375.76_{\pm 6.64}$	$1.56_{\pm 0.10}$	$1.06_{\pm 0.05}$
PROSE$_{u}$	$333.35_{\pm 13.53}$	$1.35_{\pm 0.13}$	$0.65_{\pm 0.09}$	$369.00_{\pm 21.15}$	$1.29_{\pm 0.07}$	$0.87_{\pm 0.09}$
PROSE$_{u,a}$	$304.55_{\pm 16.14}$	$1.41_{\pm 0.07}$	$0.52_{\pm 0.09}$	$352.71_{\pm 18.17}$	$1.75_{\pm 0.09}$	$1.17_{\pm 0.04}$
PROSE$_{u,a,S>1}$	$635.88_{\pm 39.19}$	$1.42_{\pm 0.04}$	$0.84_{\pm 0.05}$	$829.24_{\pm 37.09}$	$1.56_{\pm 0.10}$	$1.39_{\pm 0.08}$
PROSE$_{\text{NV}}$	$937.37_{\pm 74.59}$	$1.57_{\pm 0.05}$	$0.97_{\pm 0.11}$	$1004.21_{\pm 30.02}$	$1.36_{\pm 0.13}$	$1.32_{\pm 0.11}$
PROSE$_{\text{Full}}$	$628.28_{\pm 36.94}$	$\mathbf{1.60}_{\pm \mathbf{0.11}}$	$0.99_{\pm 0.07}$	$880.98_{\pm 22.95}$	$1.55_{\pm 0.06}$	$1.38_{\pm 0.03}$
PROSE$_{\text{Full+ICL}}$	$6912.28_{\pm 1171.86}$	$1.41_{\pm 0.10}$	$1.38_{\pm 0.05}$	7624.14 ± 973.68	$1.45_{\pm 0.13}$	$\mathbf{1.70}_{\pm \mathbf{0.07}}$

Table D.3: `Qwen2.5-72b-Instruct` + PROSE's performance on the two tasks measured by the quality of inferred preferences (Pref. Sim.) and preference compliance (PPCM) compared against no preference conditioning (NPC), true preference conditioning (Oracle), in-context learning (ICL), CIPHER [109], and ablations over PROSE's components. Results are the mean and standard error across five seeds. Best non-Oracle results per task are bolded.

	Summarization			Emails		
Method	Pref Len	Pref. Sim.	PPCM	Pref Len	Pref. Sim.	PPCM
No Learning Baselines						
NPC	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$-1.05_{\pm 0.02}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$-0.91_{\pm 0.02}$
Oracle	$120.80_{\pm 0.00}$	$3.84_{\pm 0.08}$	$1.70_{\pm 0.04}$	$118.50_{\pm 0.00}$	$3.95_{\pm 0.05}$	$1.93_{\pm 0.01}$
Learning Baselines						
ICL	$6237.54_{\pm 1163.61}$	$0.00_{\pm 0.00}$	$\mathbf{1.33}_{\pm \mathbf{0.07}}$	$6754.73_{\pm 975.36}$	$0.00_{\pm 0.00}$	$1.38_{\pm 0.08}$
CIPHER-1	$138.91_{\pm 3.67}$	$1.44_{\pm 0.03}$	$0.24_{\pm 0.07}$	$148.69_{\pm 1.51}$	$1.79_{\pm 0.03}$	$0.37_{\pm 0.05}$
CIPHER-5	$\mathbf{74.93}_{\pm \mathbf{2.42}}$	$1.26_{\pm 0.10}$	$-0.05_{\pm 0.08}$	$\mathbf{78.09}_{\pm \mathbf{2.17}}$	$\mathbf{1.74}_{\pm \mathbf{0.08}}$	$0.30_{\pm 0.10}$
PROSE Ablations						
PROSE$_{\text{CE}}$	$384.12_{\pm 12.65}$	$1.44_{\pm 0.07}$	$0.67_{\pm 0.05}$	$412.09_{\pm 12.57}$	$1.57_{\pm 0.03}$	$1.03_{\pm 0.08}$
PROSE$_{u}$	$254.81_{\pm 15.59}$	$1.55_{\pm 0.12}$	$0.47_{\pm 0.11}$	$364.86_{\pm 13.18}$	$1.40_{\pm 0.10}$	$1.01_{\pm 0.10}$
PROSE$_{u,a}$	$321.09_{\pm 8.23}$	$\mathbf{1.76}_{\pm \mathbf{0.11}}$	$0.70_{\pm 0.07}$	$375.89_{\pm 24.73}$	$1.75_{\pm 0.08}$	$1.24_{\pm 0.05}$
PROSE$_{u,a,S>1}$	$551.56_{\pm 16.27}$	$1.46_{\pm 0.11}$	$0.82_{\pm 0.13}$	$745.58_{\pm 29.95}$	$1.50_{\pm 0.08}$	$1.30_{\pm 0.13}$
PROSE$_{\text{NV}}$	$699.89_{\pm 22.87}$	$1.48_{\pm 0.07}$	$0.94_{\pm 0.05}$	$798.89_{\pm 37.11}$	$1.30_{\pm 0.10}$	$1.16_{\pm 0.07}$
PROSE$_{\text{Full}}$	$575.88_{\pm 20.20}$	$1.58_{\pm 0.08}$	$0.91_{\pm 0.06}$	$699.25_{\pm 33.42}$	$1.60_{\pm 0.07}$	$1.25_{\pm 0.09}$
PROSE$_{\text{Full+ICL}}$	$6795.69_{\pm 1167.68}$	$1.42_{\pm 0.08}$	$1.27_{\pm 0.09}$	$7542.95_{\pm 977.22}$	$1.52_{\pm 0.08}$	$\mathbf{1.70}_{\pm \mathbf{0.04}}$

Table D.4: `GPT-4o-mini` PROSE's performance on the two tasks measured by the quality of inferred preferences (Pref. Sim.) and preference compliance (PPCM) compared against no preference conditioned (NPC), true preference generation (Oracle), in-context learning (ICL), CIPHER [109], and ablations over PROSE's components. Results are the mean and standard error across five seeds. Best non-Oracle results per task are bolded.

Method	Summarization Pref Len	Summarization Pref. Sim.	Summarization PPCM	Emails Pref Len	Emails Pref. Sim.	Emails PPCM
No Learning Baselines						
NPC	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$-1.06_{\pm 0.04}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$-0.88_{\pm 0.04}$
Oracle	$120.80_{\pm 0.00}$	$3.84_{\pm 0.08}$	$1.69_{\pm 0.04}$	$118.50_{\pm 0.00}$	$3.85_{\pm 0.06}$	$1.97_{\pm 0.01}$
Learning Baselines						
ICL	$6237.54_{\pm 1163.61}$	$0.00_{\pm 0.00}$	$1.24_{\pm 0.09}$	$6754.73_{\pm 975.36}$	$0.00_{\pm 0.00}$	$1.40_{\pm 0.04}$
CIPHER-1	$60.80_{\pm 1.24}$	$1.23_{\pm 0.04}$	$-0.14_{\pm 0.07}$	$67.62_{\pm 1.37}$	$1.75_{\pm 0.07}$	$0.41_{\pm 0.06}$
CIPHER-5	$\mathbf{56.03}_{\pm \mathbf{1.52}}$	$1.43_{\pm 0.07}$	$-0.11_{\pm 0.13}$	$\mathbf{56.81}_{\pm \mathbf{2.00}}$	$1.76_{\pm 0.05}$	$0.38_{\pm 0.06}$
PROSE Ablations						
PROSE_{CE}	$314.11_{\pm 19.75}$	$1.63_{\pm 0.05}$	$0.66_{\pm 0.11}$	$342.27_{\pm 12.98}$	$1.71_{\pm 0.08}$	$1.18_{\pm 0.10}$
PROSE_{u}	$228.11_{\pm 11.47}$	$1.84_{\pm 0.10}$	$0.77_{\pm 0.14}$	$303.74_{\pm 7.32}$	$1.84_{\pm 0.14}$	$1.30_{\pm 0.06}$
$\text{PROSE}_{u,a}$	$249.26_{\pm 11.70}$	$1.62_{\pm 0.12}$	$0.70_{\pm 0.10}$	$317.54_{\pm 6.48}$	$1.82_{\pm 0.09}$	$1.29_{\pm 0.10}$
$\text{PROSE}_{u,a,S>1}$	$428.30_{\pm 18.53}$	$1.83_{\pm 0.17}$	$0.92_{\pm 0.08}$	$534.29_{\pm 18.92}$	$1.79_{\pm 0.06}$	$1.33_{\pm 0.01}$
PROSE_{NV}	$489.66_{\pm 13.24}$	$1.94_{\pm 0.06}$	$1.07_{\pm 0.12}$	$513.60_{\pm 11.57}$	$\mathbf{1.90}_{\pm \mathbf{0.05}}$	$1.50_{\pm 0.07}$
$\text{PROSE}_{\text{Full}}$	$446.73_{\pm 10.71}$	$\mathbf{1.98}_{\pm \mathbf{0.06}}$	$1.15_{\pm 0.07}$	$532.34_{\pm 7.78}$	$1.83_{\pm 0.08}$	$1.55_{\pm 0.03}$
$\text{PROSE}_{\text{Full+ICL}}$	$6719.98_{\pm 1164.23}$	$1.77_{\pm 0.10}$	$\mathbf{1.46}_{\pm \mathbf{0.05}}$	$7297.62_{\pm 980.54}$	$1.64_{\pm 0.05}$	$\mathbf{1.67}_{\pm \mathbf{0.04}}$

Table D.5: `GPT-4o` + PROSE's performance on the two writing tasks measured by the correctness of inferred preferences (Pref. Sim.) and preference compliance (PPCM) compared against no-preference conditioning (NPC), true preference generation (Oracle), in-context learning (ICL), CIPHER [109], and ablations over PROSE's components. Results are the mean and standard error across five seeds. Best non-Oracle results per task are bolded.

D.3.3 PRELUDE Results

Results on PRELUDE [109] for PROSE and baselines: a no-preference conditioning (NPC), an oracle preference baseline, in-context learning (ICL), CIPHER-1, and CIPHER-5 [109] (Table D.6). To directly evaluate the ability to infer preferences, we provide all models with ground-truth knowledge of the source of the documents. On the summarization task, PROSE outperforms all baselines on action/generation quality. On the email writing task, PROSE outperforms all baselines on the PPCM metric, but slightly underperforms CIPHER-1 on the poorly correlated Levenshtein distance metric (see Section 5.1.4-**Metric Correlation** for issues with Levenshtein distance).

Results in this table further support issues with the current preference-quality metrics. In the email writing task, the no-learning baseline (which always uses an empty preference), has a higher accuracy than any learning method, which may be due to the significant overlap between preference sets in the task. Further, in both tasks, the highest preference-quality scores do not lead to the highest action-quality scores. We encourage future work to look into alternative preference-quality metrics.

We lastly note that PRELUDE has substantially smaller range between the no-preference conditioned (NPC) and oracle preference baselines relative to PLUME. On PPCM, PRELUDE has a range 2.45 and 0.62 for summarization and email writing respectively, while PLUME has ranges of 3.17 and 2.91 for the two tasks. This further supports PLUME as the primary evaluation environment.

Summarization				
Method	Accuracy	BScore	Levenshtein	PPCM
No Learning Baselines				
NPC	$0.20_{\pm 0.00}$	$-0.43_{\pm 0.00}$	$107.80_{\pm 6.04}$	$-0.74_{\pm 0.10}$
Oracle	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.08_{\pm 1.48}$	$1.62_{\pm 0.09}$
Learning Baselines				
ICL	$0.20_{\pm 0.00}$	$-0.43_{\pm 0.00}$	$104.24_{\pm 8.85}$	$-0.71_{\pm 0.19}$
CIPHER-1	$0.61_{\pm 0.06}$	$0.13_{\pm 0.03}$	$48.01_{\pm 10.76}$	$0.74_{\pm 0.24}$
CIPHER-5	$0.46_{\pm 0.02}$	$0.02_{\pm 0.01}$	$33.86_{\pm 19.89}$	$0.82_{\pm 0.50}$
PROSE	$0.68_{\pm 0.12}$	$0.01_{\pm 0.03}$	$9.30_{\pm 8.70}$	$1.18_{\pm 0.16}$

Emails				
Method	Accuracy	BScore	Levenshtein	PPCM
No Learning Baselines				
NPC	$0.25_{\pm 0.00}$	$-0.37_{\pm 0.00}$	$48.12_{\pm 11.44}$	$0.86_{\pm 0.08}$
Oracle	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.02_{\pm 2.03}$	$1.57_{\pm 0.13}$
Learning Baselines				
ICL	$0.25_{\pm 0.00}$	$-0.37_{\pm 0.00}$	$53.38_{\pm 13.46}$	$0.87_{\pm 0.13}$
CIPHER-1	$0.03_{\pm 0.06}$	$-0.16_{\pm 0.04}$	$12.06_{\pm 15.04}$	$1.04_{\pm 0.12}$
CIPHER-5	$0.25_{\pm 0.00}$	$-0.08_{\pm 0.04}$	$13.34_{\pm 9.57}$	$1.09_{\pm 0.08}$
PROSE	$0.01_{\pm 0.03}$	$-0.22_{\pm 0.02}$	$14.05_{\pm 6.14}$	$1.10_{\pm 0.06}$

Table D.6: **PRELUDE Results**. PROSE's ability to infer the correct preference set and generation quality across the two PRELUDE tasks compared against a no-preference conditioning baseline (NPC), a method with access to the true preferences (Oracle), in-context learning (ICL), and CIPHER [109]. Results are reported as the mean and standard error across five seeds. Accuracy and Bscore (BERTScore) [419] are preference-quality metrics, while Levenshtein distance and PPCM (per preference-component match) are action-quality metrics.

D.3.4 Preference Inference and Conditioning Performance by Number of User Samples

In Figure D.3 we show the impact of the number of samples for a given user according to the measures for inferred-preference and generation quality metrics.

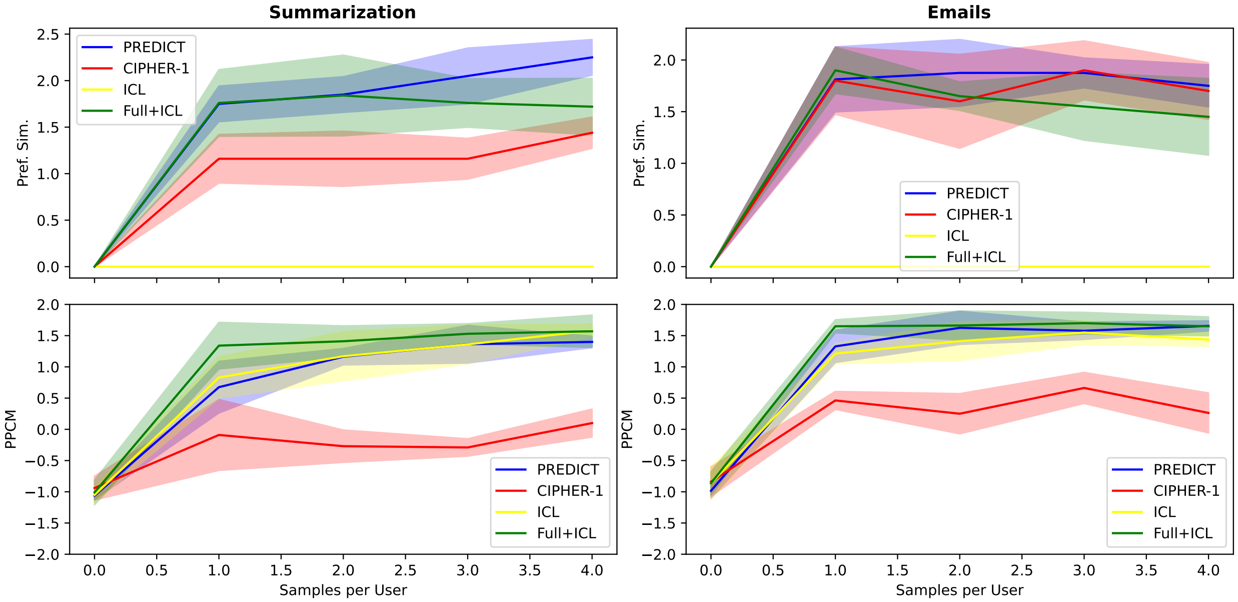


Figure D.3: Performance for PROSE, CIPHER-1, in-context learning (ICL), and PROSE+ICL given different numbers of user samples to learn from. Mean and standard error over five seeds for preference quality (Pref. Sim.) and preference-conditioned generation quality (PPCM). `GPT-4o` is the LLM.

D.4 PRELUDE vs. PLUME Preference Sets

The preference sets used for each document source and environment (PRELUDE vs. PLUME) are given in Table D.7.

Document Source	Task Version	User Preferences
		Summarization
News Articles	PRELUDE	interactive, playful language, positive, short sentences, storytelling, style targeted to young children
	PLUME	adopt a step-by-step structure, include a simile, use ampersands (&) instead of "and"s, write in the style of a children's book
Chat Forum Posts	PRELUDE	brief, immersive, invoke personal reflection, second person narrative, show emotions
	PLUME	adopt a header and sub-header structure, include rhetorical questions, use ALLCAPS to emphasize words, write in the style of a tweet
Encylopedia Pages	PRELUDE	brief, bullet points, parallel structure
	PLUME	adopt a rhyming structure, include modern slang, use semicolons (;) when possible, write in the style of a screenplay
Paper Abstract	PRELUDE	inquisitive, simple English, skillful foreshadowing, tweet style, with emojis
	PLUME	adopt a question-answering style structure, include personifications, use archaic language, write in the style of a podcast
Movie Review	PRELUDE	question answering style
	PLUME	adopt a stream-of-consciousness structure, include onomatopoeias, use imagery, write in the style of old timey radio
		Email Writing
Personal Problem	PRELUDE	conversational, informal, no closing
	PLUME	be intensely emotional, include alliterations, use a formal tone, write in a second person narrative
Paper Review	PRELUDE	call to action, casual tone, clear, positive
	PLUME	be sharply critical, include several short and punchy sentences, use parenthetical asides, write using assertive expressions
Paper Tweet	PRELUDE	engaging, personalized, professional tone, thankful closing
	PLUME	be blatantly sarcastic, include hyperboles, use an informal tone, write in a third person perspective
Paper Summary	PRELUDE	professional greeting and closing, respectful, straight to the points, structured
	PLUME	be highly inquisitive, include several long and flowing sentences, use emojis, write using conditional expressions

Table D.7: The user preferences for each assistive writing task (summarization vs. email writing), task topic, and benchmark version.

D.5 Illustrative Examples of Issues with PRELUDE

D.5.1 Levenshtein Distance and Multiple Generations

We present here an inherent issue with Levenshtein distance. For all three generations, we prompt gpt-4o to summarize the encyclopedia article on hotdogs. The first two generations (A. and B.) use identical prompts, in which we ask it to use the preference: "use a haiku". In the third generation, we ask it to use the preference "use a single short sentence". We then calculate the Levenshtein distance between each pair, reported in Table D.8. Notably, the Levenshtein distance and length normalized Levenshtein distance, in which a lower score represents higher similarity, would both indicate that generation A and C are more similar to generation A and B, even though generation A and B share the same exact same prompt while generation C uses a entirely different preference. BERTScore does a better job of identifying the most similar texts, however the PPCM scores, which compares writing directly to the true preferences, clearly labels both Generation A. and Generation B. as "strongly exhibiting" the preference "use a haiku", and Generation C. as "strongly contradicting" the preference.

Generation A.

A sausage in bun,
Toppings vary near and far,
Hot dog, rich in roots.

Generation B.

Grilled or steamed sausage,
In a bun with cheese or spice,
Street food's warm embrace.

Generation C.

A hot dog is a sausage in a bun with various toppings such as mustard, ketchup, and relish

Pairing	Levenshtein Dist.	Norm. Levenshtein Dist.	BERTScore	PPCM per generation
A-B	17	0.85	0.27	(2, 2)
A-C	16	0.76	0.20	(2, -2)
B-C	17	0.81	0.10	(2, -2)

Table D.8: Levenshtein distance between each pair of generations.

D.5.2 Editing Influences the User

In this next example, we demonstrate how the editing procedure can influence the output of the user. We again ask an LLM to summarize the encyclopedia article on hot dog. Let's assume, an inferring agent has incorrectly inferred the preference "question answering style". Using this preference, the preference-conditioned agent writes:

> What is a hot dog?
>
> A hot dog is a dish made of a sausage (typically a wiener or frankfurter) served in a partially sliced bun. It can refer to the sausage itself, which is grilled, steamed, or boiled.
>
> How is a hot dog typically prepared?
>
> Hot dog preparation varies, but common condiments include mustard, ketchup, relish, and onions. Some toppings can also include cheese sauce, sauerkraut, chili, bacon, or jalapeños.
>
> Where did hot dogs originate?
>
> The sausages used in hot dogs were culturally imported from Germany, and the dish became popular in the U.S. as a working-class street food. Over time, hot dogs became closely associated with American culture, particularly baseball.
>
> Where are hot dogs most popular?
>
> While they are especially associated with New York City, hot dogs became widespread across the United States throughout the 20th century, becoming an integral part of American cuisine.

We ask the user if the above is satisfactory for it's true preference of: "style targeted to young children". It does not, so we then ask the user to edit the above for it true preference "style targeted to young children", and the user writes:

> What is a hot dog?
>
> A hot dog is a yummy snack made of a sausage (like a wiener or frankfurter) inside a soft bun. Sometimes people call the sausage itself a hot dog, and it can be cooked by grilling, steaming, or boiling.
>
> How is a hot dog usually made?
>
> Hot dogs can be made in different ways, but they often have yummy toppings like mustard, ketchup, relish, and onions. Some people like to add even more fun stuff like cheese, sauerkraut, chili, bacon, or spicy jalapeños.
>
> Where did hot dogs come from?
>
> The sausages in hot dogs came from Germany, and a long time ago, they became super popular in the U.S. as a quick and tasty street food for busy people. Now, hot dogs are a big part of American culture, especially at baseball games!

> Where are hot dogs the most popular?
>
> Even though hot dogs are popular all over, they are especially loved in New York. Over the years, hot dogs have become a favorite food all across the United States!

While the above summary does correctly follow the true preference, it is still formatted in a question answering style, which is not part of the user's preferences. I.e., the preference-inferring agent's error has directly impacted the user's writing. This is another reason why PLUME removes the editing procedure.

For reference, if the user is directly asked to summmarize the encyclopedia article on hot dogs using it's true preference of "style targeted to young children", it outputs:

> A hot dog is a yummy food that's made with a sausage, like a wiener or frankfurter, inside a soft bun. You can cook the sausage in different ways, like grilling or boiling. People love to put tasty toppings on their hot dogs, such as ketchup, mustard, cheese, and even onions!
>
> Hot dogs came from Germany, but they became super popular in the US, especially at baseball games. Now, you can find hot dogs all over the country, and in some places, like Chicago, people have special ways of making them!

D.6 Prompts

D.6.1 Preference Inference and Preference-Conditioned Agent Prompts

The prompts used by PROSE for writing sample generation and task completion in the PLUME environment are in Appendix Figure D.4. The prompts used by PROSE to infer the user preferences from user examples are provided in Appendix Figure D.5 for the PLUME environment.

System Prompt

You are an experienced writer. Adapt your writing to heavily emphasize the provided preferences.

User Prompt

You have the following preferences: `<preference_description>`

Using these preferences, write a short {`summary` | `email`} about {`this` | `these`} {`article` | `notes`}:

[START OF {`ARTICLE` | `NOTES`}]
`<task_content>`
[END OF {`ARTICLE` | `NOTES`}]

Encapsulate the {`summary` | `email`} in triple quotes
“““
<{`summary` | `email`}>
”””

Figure D.4: LLM prompts for the **preference-conditioned agent** and for **task completion** on the **PLUME’s** summarization and e-mail writing tasks. The system prompt is prepended to the user prompt following the LLM’s chat template. “{...|...}” means that of the two options is selected based on the task and “`<...>`” indicates that the text is formatted from a variable. `inferred_preference_i` refers to one of the inferred user preferences.

System Prompt

A user is completing writing tasks. The user has an underlying set of preferences that explains why they write the way they do.

User Prompt

Aggregation Task

We are tasked to curate a prompt to guide a specific style of writing. We currently have the following list of preferences related to writing styles:
[`<preference_description` ,..., `<inferred_preference_`l`>`]
Unfortunately, these preferences may overlap or contain redundancies. Please review the list and condense it by combining similar or overlapping preferences, ensuring that the distinct intent behind each one remains clear so that a writer can easily follow them. Ensure the condensed list is concise, non-redundant, and preserves the original level of specificity. When applicable, preserve the exact wording. Return the revised preferences in the same format as the original list.

Figure D.5: LLM prompts for **preference inference** on **PLUME's** summarization and e-mail writing tasks. The system prompt is prepended to each user prompt following the LLM's chat template. "{...|...}" means that of the two options is selected based on the task and "`<...>`" indicates that the text is formatted from a variable. `user_output` refers to how the user completes the task, `assistant_output` how the assistant completes the task, and `inferred_preference_i` to one of the inferred user preferences. Continued on next page.

Inference Task

We received a new task. The task is to {`summarize`|`write an email about`} the following:
`<article|notes>`

We have previously identified the following preferences: `<preference_description>`
Based on these preferences, we wrote this {`summary`|`email`}:
`<assistant_output>`

However, this differs from the user's {`summary`|`email`}. The user wrote this {`summary`|`email`}:
`<user_output>`

Refine the list of preferences by adding, removing, or updating preferences in order to better imitate the user.

While refining the preference set, you should:
- Identify and reason about differences between our writing and the user's writing.
- Consider writing traits from distinct quirks to broader stylistic tendencies.
- Provide a concise set of preferences in the imperative form.
- Be precise; make the fewest possible changes to the preference set.
- Do not qualify, dilute, or soften existing preferences.
- Refine only the preferences if a clear difference exists. Otherwise, preserve the current preferences.

Provide a concise set of specific preferences in the imperative form. After reasoning, , output the refined set of preferences on a single new line and prefaced with "Preferences:".

Figure D.5: LLM prompts for **preference inference** on the **PLUME's** summarization and e-mail writing tasks. The system prompt is prepended to each user prompt following the LLM's chat template. "{...|...}" means that of the two options is selected based on the task and "`<...>`" indicates that the text is formatted from a variable. `user_output` refers to how the user completes the task, `assistant_output` how the assistant completes the task, and `inferred_preference_i` to one of the inferred user preferences.

User Prompt

Preference Breakdown Task

You inferred the following preference string:
`<inferred_preference_description>`
Format this preference into a concise set of preferences. Format the final set of preferences as a JSON list on a single line and prefaced with ”Preferences:”. Each element in the JSON list should be a string.The final output should look like:
Preferences: [<preference 1>,..., <preference i>, ...]

Consistency Verification Task

Validate the following preference: “`<inferred_preference_i>`” against the following writing:

`<user_output>`

Does the writing confirm or contradict the preference? Select one of the following: strongly confirms the preference, somewhat confirms the preference, is neutral toward the preference, somewhat contradicts the preference, strongly contradicts the preference. Your final decision should be output on a separate line prefaced with “Verdict:”.

Figure D.5: LLM prompts for **preference inference** on the **PLUME’s** summarization and e-mail writing tasks. The system prompt is prepended to each user prompt following the LLM’s chat template. “{...|...}” means that of the two options is selected based on the task and “`<...>`” indicates that the text is formatted from a variable. `user_output` refers to how the user completes the task, `assistant_output` how the assistant completes the task, and `inferred_preference_i` to one of the inferred user preferences.

D.6.2 Synthetic Human Prompts

The prompts used to have GPT-4o play the role of our synthetic human for PROSE are given in Appendix Figure D.6. The “human” is instructed to complete the task in the same way as the preference-conditioned agent when completing the writing tasks (see Appendix Figure D.4).

System Prompt

You are an experienced writer. Adapt your writing to heavily emphasize the provided preferences.

User Prompt

You have the following preferences: `<ground_truth_preference_description>`

Using these preferences, write a short {`summary` | `email`} about {`this` | `these`} {`article` | `notes`}:

[START OF {`ARTICLE` | `NOTES`}]
`<task_content>`
[END OF {`ARTICLE` | `NOTES`}]

Encapsulate the {`summary` | `email`} in triple quotes
“““
`<{summary | email}>`
”””

Figure D.6: LLM prompts for the **synthetic human** on the **PLUME’s** summarization and e-mail writing tasks. The system prompt is prepended to the user prompt following the LLM’s chat template. “{...|...}” means that of the two options is selected based on the task and “`<...>`” indicates that the text is formatted from a variable. `inferred_preference_i` refers to one of the inferred user preferences.

D.6.3 Preference-Conditioned Agent Baseline Prompts

The prompts used in the no-preference baseline are in Appendix Figure D.7 and for the in-context learning baseline are in Appendix Figure D.8. For the in-context learning baseline, the number of examples l matches the number of examples used when coalescing prevoiusly inferred prompts (see Appendix Figure D.5).

System Prompt

You are an experienced writer.

User Prompt

Write a short {`summary` | `email`} about {`this` | `these`} {`article` | `notes`}:

[START OF {`ARTICLE` | `NOTES`}]
`<task_content>`
[END OF {`ARTICLE` | `NOTES`}]

Figure D.7: LLM prompts for the **no preference baseline** in the **PLUME environment**. The system prompt is prepended to the user prompt following the LLM's chat template. "`<...>`" indicates that the text is formatted from a variable. `task_content` refers to the content of either the article to be summarized or the notes to include in the email, depending on the sub-task.

System Prompt

You are an experienced writer. Adapt your writing to heavily emphasize the provided preferences.

User Prompt

You have previously observed the following examples:

Example 0:
{`Article` | `Notes`}:
[START OF {`ARTICLE` | `NOTES`}]
<`task_content`>
[END OF {`ARTICLE` | `NOTES`}]

{`Article` | `Notes`}:
“““
<`completion_0`>
”””

...

Example l:
{`Article` | `Notes`}:
[START OF {`ARTICLE` | `NOTES`}]
<`task_content`>
[END OF {`ARTICLE` | `NOTES`}]

{`Article` | `Notes`}:
“““
<`completion_`l>
”””

Using the same style as these examples, write a short {`summary` | `email`} about {`this` | `these`} {`article` | `notes`}:

[START OF {`ARTICLE` | `NOTES`}]
<`task_content`>
[END OF {`ARTICLE` | `NOTES`}]

Encapsulate the {`summary` | `email`} in triple quotes
“““
<{`summary` | `email`}>
”””

Figure D.8: LLM prompts for the **in-context learning baseline** in the **PLUME environment**. The system prompt is prepended to the user prompt following the LLM's chat template. “<`...`>” indicates that the text is formatted from a variable, and `completion_`l refers to an example completion provided for in-context learning. `task_content` refers to the content of either the article to be summarized or the notes to include in the email, depending on the sub-task.

D.6.4 Qualitative Verification Consistency Examples

Preference components after refinement	**`Most relevant true preference` OR *Notes***
Write in a whimsical, playful, and narrative style using vivid and childlike imagery	`Write in the style of a children's book`
Maintain a hopeful, conversational, and informal tone	`Write in the style of a children's book`
~~Mention geographical context early in a simple manner~~	*Overfit to a specific example*
Focus on the sequence of events with explicitly numbered steps labeled as 'first,' 'next,' 'then,' 'after that,' and 'finally'	`Adopt a step-by-step structure`
Use simple and metaphorical language for emotional aspects	`Include a simile`
~~Include personal details about characters~~	*Overfit to a specific example*
Use ampersands for conjunctions	`Use ampersands (&) instead of ``and''s`
Conclude with a practical reminder or lesson emphasizing support and teamwork	`Write in the style of a children's book`
~~Use rhetorical questions sparingly~~	*Discarded because they were used more than sparingly*
Capitalize key traits/actions for emphasis	`Use ALLCAPS to emphasize certain words`
Use third-person perspective	`Adopt a third-person narrative`
Use emojis strategically	`Write in the style of a tweet`
Use informal and playful language	`Write in the style of a tweet`
Use hashtags to encapsulate themes	`Write in the style of a tweet`
~~Minimize emotionally charged phrases~~	*Irrelevant*
Use direct questions	`Include rhetorical questions`
~~Be concise and direct~~	*Irrelevant*
Focus on emotional impact and highlight key themes	*Incorrect, but related to the content of many chat forum posts*
Highlight internal conflict	*Incorrect, but related to the content of many chat forum posts*
use direct address with a casual and contemporary tone	`Include modern slang`
include character interactions with informal dialogue and consistent rhyming couplets	`Adopt a rhyming structure`
employ a simplified screenplay format focusing on dialogue and voiceover	`write in the style of a screenplay`
incorporate slang and playful language	`Include modern slang`
highlight excitement and stakes dynamically	*Loosely related to screenplay*
use thematic transitions between sections	`Write in the style of a screenplay`
~~include metric prefixes explicitly~~	*Overfit to a specific example*

Table D.9: Qualitative examples of Verification. ~~Strikethrough~~ indicates the preference component was pruned. Verification successfully removes overfit of irrelevant preferences. On occasion, it discards relevant, but misqualified components.

D.6.5 Qualitative Iterative Refinement Examples

Refinement step	Inferred preference descriptions
True preferences	adopt a third person narrative, include rhetorical questions, use ALLCAPS to emphasize certain words, write in the style of a tweet
1	**Use rhetorical questions, capitalize for emphasis, be concise, include symbols or emojis, focus on emotional impact.**
2	Use rhetorical questions, capitalize for emphasis, be concise, include symbols or emojis, focus on emotional impact, **use hashtags, use symbols for brevity.**
3	Use rhetorical **questions strategically**, capitalize for emphasis, be concise, **limit** symbols or emojis, focus on emotional impact, **use 1-2 hashtags**, use symbols for brevity.
4	Use rhetorical questions strategically, capitalize for emphasis, be concise, limit **symbols**, focus on emotional impact, use 1-2 hashtags, use symbols for brevity, **incorporate emojis for emphasis.**
5	Use rhetorical questions strategically, capitalize for emphasis, be concise, **use ”&” for brevity**, focus on emotional impact, use 1-2 hashtags, **use fewer emojis for emphasis, highlight key themes.**
True preferences	adopt a question-answering style structure, include personifications, use archaic language, write in the style of a podcast
1	**Use a poetic and narrative style with vivid imagery and metaphor; employ archaic language and a conversational tone; structure writing like a narrative or script; directly address the audience to enhance engagement.**
2	Use a **podcast or broadcast format** with vivid storytelling imagery and metaphor; employ **slightly modern** archaic language and a conversational tone; structure writing **as a continuous narrative**; directly address the audience to enhance engagement.
3	Use a podcast or broadcast format with **named episodes**; employ **consistently** archaic language **with varied vocabulary**; **use thematic and specific metaphors; enhance audience engagement with direct questions and conversational elements**; structure writing **with a clear narrative arc and defined sections**.
4	Use a podcast or broadcast format with **creatively thematic episode titles**; employ consistently archaic **and poetic** language with varied vocabulary; **use vivid and personified metaphors**; enhance audience engagement with **rhetorical style and subtle questions**; structure writing with a clear narrative arc and **poetic conclusions.**
5	Use a podcast or broadcast format with creatively thematic episode **titles evoking transformation and mystery**; employ **consistently archaic** language throughout; use vivid and personified metaphors; enhance audience engagement with **frequent** rhetorical questions; structure writing with a clear narrative arc and **conclude with harmony and enlightenment.**
True preferences	adopt a second person narrative, include onomatopoeias, use imagery, write in the style of old timey radio
1	**Use vivid imagery and metaphor, adopt a narrative style, address the reader directly, and create an immersive experience.**
2	Use vivid imagery and metaphor, adopt a narrative style, address the reader directly **with a conversational and auditory tone**, create an immersive **and nostalgic** experience.
3	Use vivid imagery and metaphor, adopt a narrative style, address the reader directly with a conversational and auditory tone, create an immersive **experience with a focus on auditory imagery and live storytelling.**
4	Use vivid **auditory** imagery and metaphor, adopt a narrative **style reminiscent of a radio broadcast**, address the reader directly with a conversational and auditory tone, create an immersive experience with a focus on **auditory** storytelling.
5	Use vivid auditory imagery and metaphor **with a nostalgic and whimsical tone**, adopt a narrative style reminiscent of a classic radio broadcast, address the reader directly with a conversational and auditory tone, create an immersive experience with a focus on auditory storytelling.

Table D.10: Qualitative examples of Iterative Refinement. Bold indicates modifications performed by the refinement step. Note: the true preferences are for reference only, the initial refinement step is conditioned on an empty preference description.