Unifying Foundation Models with Decision Making: A Path Towards Autonomous Self-Improving Agents beyond Rewards and Human Supervision

by

Abhranil Chandra

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2025

© Abhranil Chandra 2025

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Recent progress in AI has primarily been driven along two axes - scaling large generative Foundation Models (FMs) trained on static internet scale data and developing better sequential decision making (DM) and reinforcement learning (RL) algorithms that allow experiential learning. Both paradigms alone have their own flaws, but together provide a scalable recipe towards general intelligence.

As AlphaGo made its famous "Move 37", it reaffirmed Reinforcement Learning's(RL) efficacy as a paradigm to bootstrap intelligence via interactive learning from scratch, optimizing for goal-directed behavior, self-improvement, and emergence of novel superhuman abilities. However, RL is hard to scale beyond narrow tasks or simulated environments given the scarcity of real-world decision-centric data, sparsity of feedback, hard-to-design rewards, and problems of scaling to larger models. On the contrary, recent generative FMs pretrained on static internet-scale text and image data excel at broad high-level knowledge acquisition but fail to acquire robust internal World Models, precluding their notion of agency, ability to plan and reason well, and extrapolation beyond training data.

In the thesis we focused on combining these complementary paradigms - to both build and use FMs for DM and develop DM and RL tools for improving FMs - using DM/RL as a paradigm to finetune and optimize general-purpose pretrained models to elicit better decisions beyond training data, rather than viewing it as a paradigm to bootstrap intelligence from scratch. Such broadly capable systems can then be used to empower agents that can perceive, reason, and act robustly in both physical settings to complete embodied tasks and in virtual settings to help as autonomous task completion and knowledge-discovery agents.

First, we introduce *VideoAgent*, a jointly trained goal-conditioned video generation policy and self-improving simulator for embodied planning tasks. VideoAgent learns to refine its own generated plans based on a novel self-conditioning consistency objective and also using feedback from pretrained vision-language models (VLMs), without requiring ground-truth action labels or any explicit rewards. The model further leverages search to enable iterative improvement of the video plans using inference-time compute, leading to more grounded and physically plausible plans in robotic manipulation tasks.

Second, we develop a framework, *Reg-ReBoot*, to investigate efficient and scalable methods to improve base non-reasoning LLMs into better reasoners without using explicit verified data or rule-based verifiers. We analyze this counterintuitive idea: fine-tuning language models using unverified and even incorrect reasoning traces leads to reasoning improvement. We show that large language models (LLMs), due to their inductive biases, can learn useful reasoning heuristics by averaging over noisy chain-of-thought (CoT) data.

Our results on mathematical reasoning benchmarks reveal that noisy synthetic data can be an efficient way to bootstrap performance and decrease reliance on hard to curate verified solutions. From these insights we propose a two stage mid-training pipeline - lowering the barrier to scalable reasoning improvement.

Finally, we address the evaluation bottleneck in generative modeling by proposing *ReFeR*, a multi-agent, tuning-free evaluation framework. ReFeR uses a hierarchy of pretrained LLMs and VLMs to provide automatic, scalable, and high-quality feedback for both textual and visual generations. This framework not only rivals human-level evaluation accuracy but also produces structured feedback, enabling downstream distillation and fine-tuning, and proving effective even in complex reasoning tasks.

Together, through these works, I try to contribute towards the goal of building autonomous self-improving agents: one where systems powered by foundation models leverage test-time compute, generative simulations and world models, and diverse learning signals beyond explicit rewards and human feedback to drive interactive learning and decision making.

Keywords: Machine Learning, Reinforcement Learning, Unsupervised Policy Learning, Generative World Models, Planning, Search, Sequential Decision Making, Video Policy, Self-Improvement, Model-Based Learning, Robot Learning, LLM, Reasoning, Scalable Oversight.

Acknowledgements

First and foremost, I would like to thank my advisor, Prof. Sebastian Fischesmiester, for being so very patient and helpful, as well as direct and honest in his guidance, and for spending so much of his time teaching me how to be a better researcher. I am also incredibly grateful for the research freedom that he gave me, to try and explore different topics and to independently shape my research direction, and this thesis. I also thank Prof. Kate Larson and Prof. Freda Shi for their valuable advice to help me improve my thesis, taking the time to be readers for my thesis and to be on my thesis committee.

Additionally, I would like to thank my wonderful research mentors during the master's program. I am incredibly lucky to have had the opportunity to work under the guidance of and learn from Dr. Sherry Yang and Dr. Bo Dai during my collaboration at Google DeepMind. I would also like to thank Prof. Aaron Courville and Dr. Rishabh Agarwal at MILA for their advice, guidance, and support during the last project of my thesis. I am also thankful to Dr. Arian Hosseini and Ayush Agrawal for helping me with this work at MILA. I am also very grateful to my undergraduate advisor, Prof. Pabitra Mitra at IIT Kharagpur, for setting me up on the path to pursue AI research, which has been, till now, one of the best decisions of my life.

I would also sincerely like to thank my friends for their constant support and motivation, even though most of them are continents apart, busy with their own hectic work and life. And to Sivangi - you made every step easier and every success sweeter. Finally and most importantly, I would like to thank my parents - they have given me the perfect, goldilocks amount of guidance throughout my life: neither so much that I relinquish ownership over my choices, nor so little that my path feels precarious and lost.

Dedication

This is dedicated to my family, friends, teachers, and mentors.

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements v

Dedication vi

List of Figures xii

List of Tables xv

1 Introduction 1

1.1 Motivation and Challenges 1

1.2 Contributions 4

1.3 Thesis Organization 6

2 Self-Improving Generative Simulators and World Models: Goal Conditioned Video Policies for Embodied Planning 8

2.1 Introduction 8

2.2 Related Work 10

2.3 Background 11

2.3.1 Video as Policy in Sequential Decision Making 12

2.3.2 Consistency Models . . . 12
2.4 Video Generation as An Agent . . . 13
2.4.1 Refinement through Self-Conditioning Consistency . . . 13
2.4.2 Inference-Time Planning through VLM-Guided Video Generation. . . 16
2.4.3 Self-Improvement through Online Finetuning . . . 17
2.4.4 Unsupervised Policy Learning via Implicit Successor Measures . . . 18
2.5 Algorithms . . . 20
2.6 Dataset Descriptions in Detail . . . 21
2.6.1 Datasets and Environments. . . . 21
2.6.2 Additional trajectories per iteration during online training . . . 23
2.7 Experiments . . . 23
2.7.1 Baselines and VideoAgent Variants. . . . 23
2.7.2 End-to-End Task Success . . . 24
2.7.3 Effect of Different Components in VideoAgent . . . 24
2.7.4 VLM Feedback for Correction . . . 28
2.7.5 Evaluating Self-Refinement on Real-World Robot Videos . . . 28
2.7.6 Details of Qualitative Evaluation on BridgeData V2 . . . 29
2.8 Extended Experiments . . . 30
2.8.1 Videos to action conversion . . . 30
2.8.2 Baseline experiments on Metaworld . . . 31
2.8.3 Further Analysis of VideoAgent-Online . . . 31
2.9 Architectural Details of VideoAgent . . . 31
2.9.1 Video Diffusion training details . . . 31
2.9.2 Inference time speed . . . 31
2.10 Prompt Structure for VLM Feedback . . . 32
2.10.1 Binary Classification . . . 32
2.10.2 Identification and Suggestion: . . . 33

2.11 Task Descriptions and In-Context Examples for VLM Feedback 40
2.12 Examples 42
2.12.1 Zero-shot generalization on real-world scenes 42
2.12.2 Improvements on real-world scenes 42
2.12.3 Improvements in Meta-World 44
2.12.4 Improvements in iThor 44
2.12.5 Identification and Suggestive Feedback Examples 46
2.13 Conclusion, Limitations and Future Work 47

3 Beyond Rule-Based and Human Verifiers: Scaling Reasoning Training Compute with Unverified Synthetic Data and Reward Free Methods 48
3.1 Introduction 48
3.2 Related Work 51
3.3 Method and Analysis: Regularized Reasoning Bootstrapping (Reg-ReBoot) 53
3.3.1 The Latent Structure of Reasoning: Primitives and Traces 53
3.3.2 Modeling Imperfect Demonstrations from a Noisy Teacher 54
3.3.3 The Implicit Regularization of Maximum Likelihood Learning . . . 55
3.3.4 Formalizing the Analogy to Cognitive Regularization 56
3.3.5 Distillation for Reasoning 57
3.4 Experiments and Results 59
3.4.1 Experimental Setup 59
3.4.2 Details of SFT 61
3.4.3 Primary Findings: Learning to Reason via Distillation and from Unverified CoTs 61
3.5 Ongoing Extensions of Experiments and Results 66
3.6 Conclusion and Future Work 69

4 Automating Evaluations for Scalable Oversight and Improving Verifications for Reasoning **71**

4.1 Introduction . . . 71
4.2 Methodology . . . 73
4.2.1 ReFeR Framework . . . 73
4.2.2 Prompting Schema . . . 74
4.2.3 ReFeR Variants . . . 75
4.3 Experiments . . . 76
4.3.1 Datasets . . . 76
4.3.2 Baselines . . . 76
4.3.3 Implementation Details . . . 78
4.4 Results and Discussions . . . 79
4.4.1 NLG Evaluation . . . 79
4.4.2 Multimodal Evaluation . . . 80
4.4.3 Reasoning . . . 81
4.5 Analysis . . . 82
4.5.1 Peer Ablation . . . 82
4.5.2 Selecting Peers and area chair . . . 84
4.5.3 Error Analysis . . . 86
4.6 Related Work . . . 88
4.7 Conclusion . . . 89

5 Conclusion **90**

5.1 Summary . . . 90
5.2 Future Work . . . 92

References **93**

APPENDICES **123**

A Appendix and Supplementary Material for Chapter 4: ReFeR **124**

A.1 Future Works . . . 124
A.2 ReFeR Algorithm . . . 126
A.3 Hyperparameters . . . 127
A.4 Prompt Optimization . . . 127
A.5 Auto Prompt . . . 129
A.6 SummEval Results . . . 132
A.7 Datasets . . . 133
A.8 Note on ReConcile . . . 134
A.9 More Related Works . . . 135
A.10 Finetuning . . . 135
A.11 Communication Strategies: . . . 136
A.12 Inference and Computation . . . 136
A.13 Statistical Tests . . . 140
A.14 Example of Reasoning improved by Framework . . . 140
A.15 Hallucination . . . 143
A.16 Homogeneous Experiments . . . 144
A.17 Prompts . . . 144
A.17.1 NLG Evaluation . . . 144
A.17.2 MultiModal Evaluation . . . 154
A.17.3 Reasoning . . . 159

List of Figures

2.1 **The VideoAgent Framework.** VideoAgent first generates a video plan conditioned on an image observation and task description similar to [50], and undergoes (1) iterative video refinement using feedback from a vision language model (VLM), (2) using the VLM to select the best refined video plan to convert to control actions through optical flow, and (3) executing the control actions in an environment and improving video generation using real-world feedback and additional data collected online. 9

2.2 **An illustration of Self-Conditioning Consistency.** The horizontal direction represents regular denoising process. The two rows represent two refinement iterations. $\hat{\mathbf{x}}_i$ denotes generated video plan at refinement iteration i. We condition the refinement iteration $i+1$ on generated video from the previous iteration $\hat{\mathbf{x}}_i$. 14

2.3 Environments and Datasets that we work with: Meta-World, iThor, and BridgeData-V2 22

2.4 **Effect of Refinement Iterations.** The accuracy of downstream tasks generally increases as the number of refinement iteration increases. 27

2.5 **Effect of Online Iterations.** The overall task success of VideoAgent increases as the number of online iterations increases. 27

2.6 Few-Shot Examples given to VLM: We provide some examples to the VLM and corresponding feedback to teach the VLM in-context how to critic the generated videos for task completion and success or failure. 41

2.7 **Zero-shot generalization of VideoAgent:** VideoAgent generalizes fairly well to natural distribution shifts and is able to generate successful trajectories on data it has not been trained on. 42

2.8 **Correcting Hallucinations in Video Generation:** The AVDC model hallucinates after the second frame, and totally removes the colander from the frames and places the banana on the table. In contrast, VideoAgent accurately retains the colander's position and correctly places the banana inside. 43

2.9 **Correcting Hallucinations in Video Generation:** The goal prompt is "Assembly" as shown in the Target Video. The AVDC model has problem of object permanence and action incomplete in last frame. In contrast, our VideoAgent model accurately object permanence and correctly places the inside the peg properly. 44

2.10 **Correcting Hallucinations in Video Generation:** The goal prompt is "Television" as shown in the Target Video, the goal is for the navigator to locate the object and reach near it. The AVDC model has difficulty reconstructing and navigating in the livingroom to find the television. In contrast, our VideoAgent model solves the initial frame hallucinations and accurately reaches near the television correctly. 45

2.11 **Detailed VLM Feedback:** We show the efficacy of VLMs to provide useful feedback even in the absence of access to a simulator or real-world execution environment. The VLM acts as a proxy reward model to condition VideoAgent on useful corrective signals, leading to improved performance as described in Table 2.3. 46

3.1 Gemma 2-2B SFT results on various datasets. The efficacy of unverified synthetic CoTs generated by a teacher model as well as self-distillation clearly helps learn reasoning skills even better than human_gold dataset. 63

3.2 Gemma 2-2B SFT results on Gemma2-27B CoTs compared to Qwen2.5-14B CoTs shows the base model's preference towards similar model family generated data. 64

3.3 Results of SFT on Llama 3.1-8B model on various datasets. Reg-ReBoot provides a good understanding of performance gains in Llama similar in trends to the Gemma model. Llama base model can bootstrap reasoning skills effectively from synthetic unverified datasets and performs significantly better than human_gold data. 65

3.4 Results of SFT on Qwen 2.5-1.5B model on various datasets. AS we can see the gains in performance are moderate compared to the other two model families given the strong base performance of Qwen on Math reasoning already. Still SFT on human_gold, and Qwen 14B distilled CoTs show some gains while the others degrade performance upon scaling data. 66

3.5 Models often prefer synthetic LLM generated data more than gold standard human data measured in terms of perplexity (lower the better). 67

4.1 **Illustration of the ReFeR Framework on the TopicalChat dataset.** Refer to Fig. A.2 (in the appendix) for illustration of ReFeR for multimodality and Algorithm 5 showing the framework's working. We use the predictions from AC to create an Instruction tuning dataset which can be used to improve the performance of smaller models as evaluators, shown in Appendix A.10 . 74

4.2 **Framework Ablation.** Results obtained on ReFeR-Pro by progressively adding different peers for the TopicalChat Dataset. The points in the figure indicate the performance of ReFeR when specific labelled peers were used in conjunction with the area chair (GPT-4o-mini). "3 Peers" refers to the Llama, Nemo, and Gemma models being used as peers. "4 Peers" includes the same 3 peers along with the Mixtral model added as fourth peer. Detailed results are presented in Table 4.5. 83

4.3 **Performance analysis wrt framework scale.** Pie-charts showing Peer and AC performance on evaluation and reasoning tasks. (P- Peer model, AC- area chair Model) 87

A.1 Prompting Schema 125

A.2 **Illustration of ReFeR for Multimodal evaluation shown on AGIQA dataset**. A similar version of ReFeR working on textual TopicalChat dataset is shown in 4.1. 125

A.3 **Ratings Distribution.** We show the distribution of human annotations for Consistency, FLuency metrics in the SummEval dataset 133

A.4 **Inference speed comparison with baselines** 137

A.5 **Performance vs Compute Cost** 139

List of Tables

2.1 **Meta-World Results.** Mean success rates of baselines and VideoAgent on 11 simulated robot manipulation environments from Meta-World. VideoAgent consistently outperforms baselines across all tasks. 25

2.2 **iThor Success Rates** comparing VideoAgent with the AVDC baseline. VideoAgent outperforms AVDC across all four rooms averaged over all three objects in each room. 26

2.3 **Effect of Different Feedback** used to train the refinement model. Descriptive feedback from the VLM leads to higher improvement in task success. . 26

2.4 **VLM Performance** measured according to whether a VLM considers a generated video as acceptable using human label as the ground truth. . . . 27

2.5 **Meta-World: VideoAgent-Feedback Guided Results** The mean success rates for various tasks, comparing different VideoAgent-Feedback Guided variants and the AVDC baseline. 34

2.6 **BridgeData-V2 Results.** Quantitative metrics comparing AVDC and VideoAgent on generated Bridge data. VideoAgent outperforms the baseline according to all except for one metric. 35

2.7 Task Success and Other Fine-grained Qualitative Evaluation Metrics on BridgeData-V2 35

2.8 **Meta-World Results.** The mean success rates of baselines and VideoAgent on 11 simulated robot manipulation environments from Meta-World. VideoAgent consistently outperforms baselines across all tasks. 36

2.9 **Meta-World Result.** The mean success rates of VideoAgent combined with successive rounds of data collection via Online Iterations and Replan modules as compared to AVDC baseline. 37

3.1 Peak SFT performance of Gemma 2-2B and Llama 3.1-8B on the MATH-128 test set. 62

3.2 Manual analysis of the quality of 100 CoTs generated by the Gemma-27B teacher model. This analysis reveals that a majority of traces leading to an incorrect final answer still contain a largely correct reasoning structure which forms the basis of our proposed method and framework Reg-ReBoot. 64

4.1 **Dataset Statistics**. We list all the tasks we tackle in our paper and the datasets we used to show results with the number of samples used.For Reasoning, a random subset of 100 was sampled from the original datasets, following [34]. 500 random samples were selected from the original AGIQA-1k to get a well-distributed dataset. We use 864 samples with usable image urls from the ICQD test dataset. We use the full test sets for the NLG Evaluation datasets. 77

4.2 **Comparison of ReFeR with baselines for NLG evaluation on the TopicalChat dataset (all baselines and ReFeR uses GPT-4o-mini as the backbone).** Results are averaged across 3 runs. The best results are bolded, and the second-best are underlined. *Costs for ReFeR Pro and ReFeR Lite include only AC API cost, as open-source peer models can be deployed locally and so do not involve API costs. Peer model costs based on API pricing from services like [207] are also provided for reference. Relative costs are shown as fractions of the most expensive method. GPT-4o-mini row denotes the performance of the AC model with the peer setup, Peer Average row denotes the correlation when the average of the 3 peer scores is considered. 80

4.3 **Multimodal Evaluation Results.** Comparison of caption quality and image generation quality score correlations with human scores on ICQD and AGIQA datasets, respectively. *X-IQE is a text-to-image VLM-based method, so we don't show it for Caption Quality. 81

4.4 **Experimental results on Reasoning tasks.** Comparison of ReFeR performance (accuracy) with single-agent and multi-agent method baselines. All results are averaged across 3 runs. Cost*- Costs are shown as relative to the most expensive method. 82

4.5 **Ablation analysis of the effect of different models used.** Comparison of ReFeR Pro results on TopicalChat with Different Peer Configurations. The method column shows what peers were used with the AreaChair (GPT-4o-mini). 4 Peers in the last rows denotes the 3 peers (Llama, Nemo, Gemma) and Mixtral. *These rows show the individual performance of the peers, not the framework's performance when the peer is used. 85

4.6 **Results on TopicalChat using the open-source model Qwen1.5-72B as the area chair.** We were unable to include results for ReFeR Pro with Qwen as the area chair due to the limitation of not being able to use n=20. 86

4.7 **Choosing model for AC**: Below Table shows individual performance of AC model and model acting as AC in the ReFeR Framework. The purpose of this table is to understand correlation of capability of the AC and our framework. We See that if gpt-4o-mini is better than gemma in base performance then gpt-4o-mini is better at being AC than gemma 87

A.1 **Analysis of Prompt Optimization.** Comparison of Average (across 4 metrics) results for different prompts on TopicalChat dataset. 128

A.2 **Analysis of Prompt Ablation.** Average results (across 4 metrics) on TopicalChat Dataset of ReFeR-Pro with different Prompt structure. 129

A.3 **Performance analysis on SummEval dataset.** Comparison of various methods for NLG evaluation on SummEval. 132

A.4 **Improving smaller models via instruction-tuning.** Finetuning Results for Mistral-7B model on TopicalChat Dataset 136

A.5 **Communication Strategies.** Results on TopicalChat showing different generation and communication strategies for ReFeR-Pro. 136

A.6 Model Comparison of FLOPs. ReFeR-Lite is calculated as the cost of the peer models(first 3 rows) and AC. ReFeR-Pro is calculated as summation of the peers and 20 AC calls. G-Eval, Analyze-Rate are calculated as 20 calls. 138

A.7 Performance and Computation of methods across different n-values. ρ is average spearman correlation across all metrics. 139

A.8 **Statistical Tests.** p-values for statistical tests between ReFeR Pro and baselines on TopicalChat. 140

A.9 Comparison of average hallucination scores across methods on the GSM8k Benchmark. GPT-4o-mini is the HHEM score of GPT-4o-mini using our peer setup. 144

A.10 Performance of ReFeR framework with same model, Gemma2-9B as both peers and AC. Varying temperatures were used for diversity among peers. . 144

Chapter 1

Introduction

1.1 Motivation and Challenges

Over the past decade, progress in AI research has been propelled along two major directions: the rise of *Foundation Models (FMs)* (like LLMs, VLMs, Vision Language Action Models and large scale Image and Video models [24]) and advances in *sequential decision making* (exemplified by superhuman reinforcement learning (RL) agents [183, 209, 20]).

Foundation models - large-scale models pretrained on broad internet data - have demonstrated exceptional capabilities in knowledge transfer across diverse vision and language tasks. For example, large language models can answer questions or generate code across domains by leveraging vast prior knowledge. However, when such pretrained models are deployed in interactive or embodied settings, they face new limitations. Unlike static prediction tasks, real-world decision making requires *agency*: the model must perceive an environment, take actions, plan over time, and adapt and self-improve. Current foundation models lack this interactive loop: they do not learn from external feedback at inference and cannot natively perform multi-step planning or maintain goals. As these models are applied to tasks involving long-term reasoning, control, and planning (e.g., dialogue or robotics), they must interface with external entities (environments, tools, or other agents) and contend with challenges like: (1) learning from feedback signals beyond supervised data (e.g., human ratings of conversation quality), (2) lacking a robust internal world models causing ungrounded and hallucinatory behaviour when applied to new environments and modalities (e.g., mapping language or vision to physical robot actions), and (3) executing long-horizon multi-turn plans after reasoning about future consequences and self-improving from learned experiences. In summary, FMs provide broad knowledge and skills, but alone

they lack the agency, adaptability, and goal-directed behavior needed for autonomous decision making.

In contrast, the field of sequential decision making using reinforcement learning has traditionally focused exactly on those interactive and experiential learning aspects: an *agent* actively interacts with an environment, receives feedback (rewards/preferences/expert demonstrations/language guidance), and improves its policy. RL algorithms have achieved remarkable feats - from surpassing human champions in Go and Atari games to enabling robots to learn complex navigation and manipulation skills. These successes demonstrate that an agent can, in principle, learn complex behaviors from scratch via trial-and-error. However, the classical RL paradigm is task-specific and data-hungry. Unlike FMs, which leverage internet-scale datasets for pretraining, RL agents typically start *tabula rasa* with limited prior knowledge. This leads to poor generalization and massive sample complexity - for instance, solving a single Atari game can require on the order of several GPU-days of experience generation. In practice, scaling up RL to real-world complexity is exceedingly difficult: each new task or environment demands extensive training, and learned policies often fail to transfer or generalize beyond their narrow training conditions. And in many cases gathering experiential data is costly or totally impossible. In summary, RL provides a framework for agency, self-improvement, autonomy, and bootstrapping emergent superhuman capabilities, but alone it struggles with efficiency, breadth of knowledge, and generalization.

These observations highlight a crucial gap - and opportunity - at the intersection of FMs and decision making algorithms. The two paradigms are highly complementary: FMs contribute general-purpose knowledge priors, representation learning, and commonsense reasoning abilities, whereas RL contributes agency, adaptation through feedback, and the machinery for sequential decision making. Unifying these strengths could enable autonomous agents that are far more powerful than either alone: agents that come equipped with broad world understanding yet can continually learn from experience, plan, and improve. Indeed, recent work has pointed out that foundation models and interactive decision making can be mutually beneficial when combined. On one hand, a pretrained model can greatly accelerate an agent's learning by providing prior knowledge and good representations for states and goals. On the other hand, adapting FMs to handle actions and feedback would allow them to operate in environments beyond their current generative abilities from static pretraining datasets, thus benefiting from interactive feedback and long-horizon planning [239, 91, 184].

However, realizing this unified vision raises several research challenges. First, we need scalable interaction paradigms that can marry the large-scale knowledge of FMs with the online learning of RL. A naive approach would be to fine-tune a giant model in an RL loop, but this may be infeasible or inefficient. An attractive alternative is to leverage

the inference-time computation of FMs to perform sophisticated reasoning and planning or search on the fly, rather than relying solely on extensive train-time adaptation. In other words, a unified agent should exploit its considerable computing power at *test time* - for example, by simulating potential futures, evaluating hypothetical actions, or doing chain-of-thought reasoning - to make better decisions without requiring billions of new training examples. Designing algorithms that effectively utilize such on-demand reasoning or search is an active area of research with many open challenges.

Second, to learn from experience at scale, an agent must be able to generate and leverage rich, informative interactions with its environment. In many domains (robotics, autonomous driving, etc.), real environment interactions are costly or risky, so a unified agent should make use of *simulated experience*. Here, generative models present a unique opportunity: a foundation model can serve as a high-fidelity simulator or world model, generating synthetic trajectories for the agent to practice with. Recent works show that powerful generative models (e.g., text-to-video models) can act as simulators of the real world, implicitly encoding physics and agent-object interactions learned from internet data. Such simulators hold the promise of internet-scale knowledge transfer - for example, a model that has watched vast amounts of human videos could simulate plausible robot experiences, helping to train a generalist robot policy. However, current generative simulators are still imperfect: they often hallucinate objects or produce physically invalid trajectories. If an agent relies on a faulty model of the world, its plans will be suboptimal or unsafe. Thus, a key challenge is enabling *self-correction and improvement* of these generative world models. The agent should not treat the simulator's predictions as ground truth, but rather critically evaluate and refine them over time.

Third, and relatedly, we must develop mechanisms for feedback and self-improvement *beyond traditional rewards*. Standard decision making and RL literature is dominated by the paradigm of using carefully designed rewards, engineered for each task. Yet in many domains, designing such complex rewards is very difficult and often impossible. Moreover, feedback in real-life scenarios can be much richer if we allow it: e.g., natural language critiques from a human or LLMs/VLMs, scores from a learned world model, preferences, or the agent's own assessment of confidence, self-critique and curiosity. Incorporating feedback beyond the traditional engineered rewards setup is essential to guide agents, using more informative dense feedback. For instance, large language models benefit enormously from human preference feedback (as seen in RLHF) and lately in reasoning from rule-based verifiers and high quality filtered data. In scenarios where human feedback or ground-truth labels or verifiable rewards are scarce, the agent should leverage *generative rewards and verification* and *intrinsic feedback*. The insight is that a models can judge its own outputs (for example, via its level of confidence or consistency) or use other models for feedback

to bootstrap a training signal. All told, achieving truly autonomous self-improvement will require agents to harness a spectrum of feedback: from traditional rewards to diverse environment feedback and preferences to their internal critiques.

In summary, bridging foundation models with decision-making algorithms opens the path toward *generally capable autonomous self-improving agents*. Such agents would combine the strengths of FMs (broad general knowledge, flexibility, rich generative modeling) with the strengths of RL (goal-directed learning, adaptability, closed-loop feedback, self-improvement). Realizing this vision demands we answer: How can a pretrained model be endowed with agency and self-learning capabilities? How can we efficiently simulate and practice complex behaviors in learned world models and simulators? How can an agent refine itself using whatever feedback is available - whether external or intrinsic?

These questions motivate the three primary works in my thesis. First, we investigate **VideoAgent**, a *self-improving generative simulator* - how an agent can use a generative world model (for example, a video predictor) not only to plan, but to *iteratively improve* its predictions via diverse feedback, and explore and collect data to further improve policy. Second, we explore **Reg-ReBoot** - reasoning skill acquisition with unverified CoT data and no reward models, exploring techniques for an agent to bootstrap complex reasoning *without* relying on ground-truth intermediate solutions or extensive human supervision. Third, we develop **ReFeR**, a *compositional system* for scalable automated evaluations of generative outputs that is *highly aligned* and calibrated with human evaluation preferences without any manual involvement. Collectively, in the thesis, I try to contribute towards the ultimate goal of unifying foundation models with decision making to build autonomous self-improving agents by tackling these three challenges mentioned above. We explore how systems powered by foundation models can leverage generative simulation, learn from diverse signals beyond explicit rewards, and use AI-driven feedback to enable robust and scalable decision-making.

1.2 Contributions

In brief, the following are contributions of my MMath thesis trying to tackle some aspects of the challenges discussed above motivated by the idea of combining the complementary paradigms of large Foundation Model priors with Decision Making and RL algorithms to make autonomous self-improving interactive agents that are less dependent on human feedback and reward specifications:

1. Video generation has been effective in generating visual plans for controlling robotic systems. Specifically, given an image observation and a language instruction, previous

work has generated video plans which are then converted to robot controls to be executed. However, a major bottleneck in leveraging video generation for control lies in the quality of the generated videos, which often suffer from hallucination (e.g., unrealistic physics), resulting in low task success when control actions are extracted from the generated videos. In this work, we propose VideoAgent to ground video generation in the physical world by integrating external feedback. VideoAgent trains a video diffusion model to perform video refinement through a novel objective which we call *self-conditioning consistency*. During inference, VideoAgent samples and refines generated video plans under the guidance of a vision-language model (VLM) as reward, enabling inference-time compute to be turned into better generated video plans. As refined video plans are executed, VideoAgent collects additional data from the environment to further improve video plan generation. Experiments in simulated robotic manipulation from MetaWorld and iTHOR show that VideoAgent drastically reduces hallucination, thereby boosting success rate of downstream manipulation tasks. We further illustrate that VideoAgent can effectively refine real-robot videos, providing an early indicator that robots can be an effective tool in grounding video generation in the physical world.

2. The prevailing paradigm for enhancing the reasoning capabilities of Large Language Models (LLMs) relies on Supervised Fine-Tuning (SFT) or Reinforcement Learning (RL) using vast datasets of human-verified filtered data or using rule-based verifiers. This approach is fundamentally limited by the prohibitive cost and logistical complexity of creating high-quality, verified data, especially for complex, multi-step reasoning tasks. In this work, we challenge this paradigm by demonstrating that LLMs can effectively learn and improve their reasoning skills by training on large-scale, unverified, and even incorrect Chain-of-Thought (CoT) examples. We introduce Regularized Reasoning Bootstrapping (Reg-ReBoot), a framework inspired by principles of children language acquisition, where learners regularize inconsistent input to extract robust underlying latent patterns. We formalize this process, showing how the standard next-token prediction objective, when applied to a dataset of imperfect CoTs, implicitly functions as a regularizer that favors consistent, correct reasoning primitives while averaging out diverse, idiosyncratic errors. Through knowledge distillation and self-distillation experiments on mathematical reasoning tasks on different non-reasoning base LLM models, we show that training on CoTs with incorrect final answers can yield performance comparable to, and in some cases exceeding, training on human-curated "gold" data. Our findings suggest a more scalable and cost-effective path forward for developing advanced reasoning in LLMs, leveraging the vast quantities of imperfect data available in the real world.

3. Assessing the quality of generative model outputs from large language models (LLMs) or vision-language models (VLMs), poses significant challenges. Traditional evaluation methods either rely on human assessment which is resource-intensive and not scalable or on automatic metrics that often correlate poorly with human preferences. Another approach is to train dedicated neural evaluators, but this typically requires substantial training data and compute. In this study, we thus introduce ReFeR, a tuning-free framework for evaluating generative outputs including both text and images, using a two-level hierarchy of pre-trained LLM and VLM evaluators. This multi-agent hierarchical strategy leverages additional compute at inference time by orchestrating multiple models and utilizing the increased test-time reasoning to boost performance. By having models themselves provide feedback and final judgments, ReFeR reduces the dependence on human evaluation. We rigorously evaluate ReFeR on four diverse evaluation benchmarks, where it surpasses prior methods in accuracy while also generating constructive feedback useful for downstream distillation and self-improvement via finetuning. Interestingly, ReFeR is also applicable for reasoning tasks - experiments on four reasoning benchmarks show ReFeR's superior collective reasoning abilities. We present two variants of the framework: ReFeR-Pro, optimized for accelerated performance, and ReFeR-Lite, offering a more test-time compute efficient solution. ReFeR-Lite is $1214\times \sim 12-14\times$ $more compute efficient than previous works while being comparably accurate to ReFeR-Pro$.

1.3 Thesis Organization

- **Chapter 1** introduces the overarching vision, motivates the research by contextualizing the complementary paradigms of foundation models and reinforcement learning, and outlines the broad challenges in this area of research and the contributions I made in the thesis in brief.
- **Chapter 2** introduces **VideoAgent**, a framework for embodied planning that leverages goal-conditioned video generation to combine learned generative self-improving world models with test-time search methods to scale robot manipulation using action and reward label free video data. This work details a novel self-conditioning consistency objective and a feedback mechanism using Vision-Language Models (VLMs) to refine generated video plans, making them more physically plausible and effective for robotic control.
- **Chapter 3** presents **Reg-ReBoot**, a framework that investigates how Large Language Models (LLMs) can learn to reason from large-scale, unverified, and even incorrect

reasoning traces. This chapter demonstrates that the inductive biases of LLMs allow them to extract robust reasoning heuristics from noisy data, thereby reducing the critical dependency on expensive, human-verified solutions and rule-based verifiers.

- **Chapter 4** introduces **ReFeR**, a multi-agent, tuning-free evaluation framework inspired by the academic peer review process to model a society of minds framework for deliberative analysis to automate evaluations. This work shows how a hierarchy of pretrained models can provide automatic, scalable, and high-quality feedback for both textual and visual generations, rivaling human-level accuracy and enabling downstream model improvement.

- **Chapter 5** concludes the thesis with a summary of our work and some directions for future work to develop generally capable autonomous and self-improving intelligent agents.

Taken together, my thesis emphasizes the central theme that progress toward generally capable autonomous agents hinges on scalable *self-improvement* and learning *beyond explicit rewards and human supervision*. Traditional supervised learning paradigms are limited by their dependence on costly, human-curated labels, while standard reinforcement learning approaches struggle with efficiency and generalization due to their reliance on carefully engineered reward specifications. By contrast, the works in my thesis illustrate a complementary pathways for overcoming these bottlenecks. VideoAgent demonstrates how generative models can act as world simulators that iteratively refine themselves using diverse feedback signals, moving beyond fixed training datasets. Reg-ReBoot shows that robust reasoning heuristics can be learned from large-scale, unverified, and even incorrect reasoning traces, thereby reducing reliance on rule-based verifiers or expensive ground-truth reasoning data. Finally, ReFeR provides a scalable evaluation framework that leverages the deliberative capabilities of foundation models themselves, reducing dependence on human evaluation and enabling iterative improvement of generative systems. Together, these directions highlight a unified perspective: foundation models with rich general priors can be transformed into interactive, self-improving agents when equipped with mechanisms to learn from data generated and collected by themselves using diverse feedback signals and their own critical evaluations beyond hand-crafted rewards and human supervision - ultimately pointing beyond the conventional boundaries of supervised and self-supervised learning learning and reward-driven reinforcement learning.

Chapter 2

Self-Improving Generative Simulators and World Models: Goal Conditioned Video Policies for Embodied Planning

2.1 Introduction

Large text-to-video (T2V) models pretrained on internet-scale data enable generation of creative video content [86, 90, 185], games [28], animations [213], and movies [267]. Recent work demonstrates their potential as real-world simulators [238, 27] and as policies with unified observation and action space [51, 109, 50]. These advances can lead to internet-scale knowledge transfer and progress toward generalist agents—a single policy for controlling multiple robots across various morphologies, environments and tasks. Nevertheless, T2V models have only had limited success in downstream applications in reality. For instance, in video generation as policy [51, 109], when an observation image and a language instruction are given to a video generation model, generated videos often hallucinate (e.g., objects randomly appear or disappear) or violate physical laws (e.g., a robot hand going through an object) [238, 27]. Such issues have led to low task success rate when generated videos are converted to control actions through inverse dynamics models, goal conditioned policies, or other action extraction mechanisms [226, 240, 5]. While scaling up dataset and model size can be effective in reducing hallucination in large language models (LLMs) [89], it is more difficult in video generation as language labels for videos are labor intensive to curate and we have not yet converged to an architecture that is more favourable to scaling [240]. Scaling aside, two major directions to improve generation in LLMs have been to incorporate

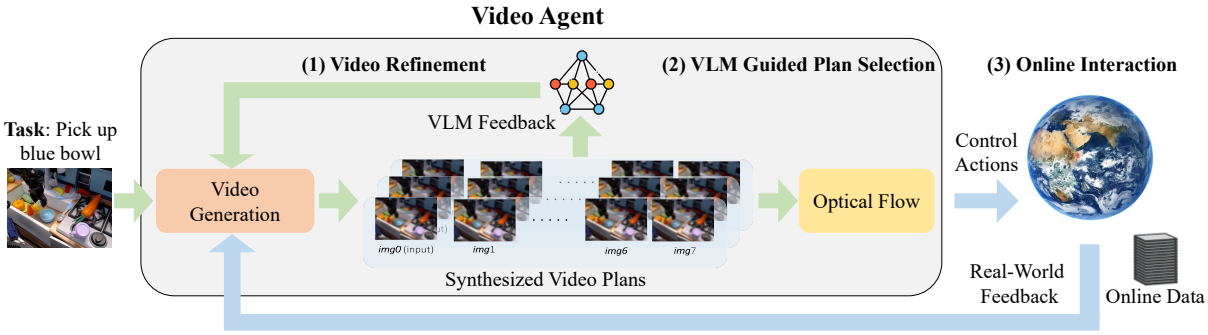


Figure 2.1: The VideoAgent Framework. VideoAgent first generates a video plan conditioned on an image observation and task description similar to [50], and undergoes (1) iterative video refinement using feedback from a vision language model (VLM), (2) using the VLM to select the best refined video plan to convert to control actions through optical flow, and (3) executing the control actions in an environment and improving video generation using real-world feedback and additional data collected online.

external feedback [156] and to scale search with increased inference time compute [187, 230]. It is therefore natural to wonder what kind of feedback and inference-time compute can be leveraged to improve T2V generations.

To answer this question, we explore two types of feedback that are natural to obtain for video generation models, namely AI feedback from a vision-language model (VLM) and real-world execution feedback when generated videos are converted to motor controls. To utilize these feedback for self-improvement, we propose VideoAgent . Different from video generation as policy, which directly turns a generated video into control actions [50, 109], VideoAgent is trained to refine a generated video plan iteratively using feedback from a pretrained VLM. During inference, VideoAgent queries the VLM to select the best refined video plan, allowing inference-time compute to be turned into better generated video plans, followed by execution of the plan in the environment. During online execution, VideoAgent observes whether the task was successfully completed, and further improves the video generation model based on the execution feedback from the environment and additional data collected from the environment. The improvement to the generated video plan comes in three folds: First, we propose *self-conditioning consistency* for video diffusion model inspired by consistency models [191, 78], which enables low-quality samples from a video diffusion model to be further refined into high-quality samples. Second, VLM feedback combined with more inference-time compute leads to better video plans. Lastly, when online access to the environment is available, VideoAgent executes the current video plan and collects additional successful trajectories to further finetune the video generation model.

A visual illustration of VideoAgent is shown in Figure 2.1.

We first evaluate VideoAgent in two simulated robotic manipulation environments, Meta-World [245] and iTHOR [113], and show that VideoAgent improves task success across all environments and tasks evaluated. We additionally provide ablation studies on the effect of different components in VideoAgent , including different types of feedback from the VLM and the amount of inference-time compute spent.Lastly, we illustrate that VideoAgent can iteratively improve real-robot videos, providing early signal that robotics can be an important mean to ground video generation models in the real world.

2.2 Related Work

Feedback and Self-improvement in LLMs. Incorporating feedback and preference signals from feedback into the finetuning process of LLMs, has led to the enormous popularity and practical usability of the current versions of LLMs as chatbots [30]. Preference feedback from humans or other AI systems [155, 120, 103] are first collected to train a reward model to guide the LLM's generation or do implicit policy optimization [177, 166]. Furthermore LLMs have shown the ability to further improve by iterative refinement during finetuning and inference [251, 247, 206]. We incorporate this reward driven improvement mechanism in our work, but unlike the LLM setting where the feedback came from a reward model or some proxy of this preference model, but in our VideoAgent work we use natural feedback from the real world when simulated videos are turned into actions that are executed in the real world.

Image and Video Generation and Editing. With the advent of large scale foundation models pretrained on internet scale data [25], generation of super realistic multimodal content has become easier. Text generation, image or video generation, and cross-modal generation [154, 169, 228, 86, 185, 237, 22] has seen major advancements leveraging the autoregressive and diffusion based models architectures. And moving beyond simple generation, these models have been leveraged for guided text, image or video editing and enhancement [93] to improve textual and visual aesthetics applied mostly to generative media [252]. But none of these existing methods focus on grounding a generative simulator in the real world to perform more complex interactive multi-turn agentic and physical tasks needing both perception and control. To solve this bottleneck, we propose VideoAgent to self-improve or edit generated plan based on grounded feedback from real-world to execute robot manipulation tasks.

Scaling Inference Time Compute. Moving beyond pretraining, scaling inference time compute has opened another axis of model performance improvement. Recent works in

inference-time compute for foundation models primarily focus on better search/planning via multiple generations and better verifiers to guide the generation process [231, 64, 126, 187]. These improvements primarily were first introduced in LLMs and have recently also been used in other foundation models like pretrained diffusion models. We also have another way to leverage inference-time compute in diffusion models by simply scaling the number of denoising steps to improve model generations [102, 189, 192]. Our work moves not only leverages the vanilla methods to leverage extra compute at inference time to refine generated video policies, but also leverage offline and online feedback to teach the model via gradient updates during finetuning to learn to self-improve.

Video Generation for Robot Learning. Leveraging videos to learn robot has been studied extensively in the research literature [148, 14, 179, 33, 158, 181, 196, 121]. Existing methods use video datasets to learn effective visual representations, to extract goal and subgoal information from videos, used as a dynamic model for planning [60, 116], or to learn via imitation from expert actions [57, 214, 138], state sequences [208, 122, 101] or action-free pretraining on videos followed by task-specific learning via RL [15, 55]. Another recent line of works has made substantial progress in leveraging generative models and video-based techniques for robotic learning and planning. One approach is to reframes sequential decision-making as a text-conditioned video generation problem, allowing policies to be inferred from synthesized video frames through inverse dynamics, dense correspondences, or by pretraining a trajectory model to predict future trajectories. [51, 109, 226]. Additionally, vision-language models in combination with text-to-video dynamics models can be used to generate long-horizon plans that can be directly executed by robots and also solve complex tasks through abstract planning grounded in visual observations [50, 5]. Moreover, generative models have been leveraged to simulate realistic agent-environment interactions, facilitating training of agents in tasks ranging from high-level planning to low-level control, enabling zero-shot deployment of agents in the real world [238]. A recent work uses test time feedback to initiate replans as needed [29]. However, these works focus mostly on replanning upon failed executions during test time, whereas our VideoAgent work improves the video generation model during traning by incorporating feedback from failed executions grounded in real-world and further refines the mistakes during test time via self-iteration and replanning.

2.3 Background

In this section, we provide the background on video generation as policy in a decision making process [50]. We also introduce consistency models [191, 78, 44], which VideoAgent builds upon for self-refinement.

2.3.1 Video as Policy in Sequential Decision Making

We consider a predictive decision process similar to [51]: $\mathcal{P} := \langle \mathcal{X}, \mathcal{G}, \mathcal{A}, H, \mathcal{E}, \mathcal{R} \rangle$, where $\mathcal{X}$ denotes an image-based observation space, $\mathcal{G}$ denotes textual task description space, $\mathcal{A}$ denotes a low-level motor control action space, and $H \in \mathbb{R}$ denotes the horizon length. We denote $\pi(\cdot|x_0, g) : \mathcal{X} \times \mathcal{G} \mapsto \Delta(\mathcal{X}^H)$[1] as the language conditioned video generation policy, which models the probability distribution over H-step image sequences $\mathbf{x} = [x_0, ..., x_H]$ determined by the first frame x_0 and the task description g. Intuitively, $\mathbf{x} \sim \pi(\cdot|x_0, g)$ correspond to possible visual paths for completing a task g. Given a sampled video plan $\mathbf{x}$, one can use a learned mapping $\rho(\cdot|\mathbf{x}) : \mathcal{X}^H \mapsto \Delta(\mathcal{A}^H)$ to extract motor controls from generated videos through a goal-conditioned policy [50], diffusion policy [21], or dense correspondence [109]. Once a sequence of motor controls $\mathbf{a} \in \mathcal{A}^H$ are extracted from the video, they are sequentially executed in the environment $\mathcal{E}$, after which a final reward $\mathcal{R} : \mathcal{A}^H \mapsto \{0, 1\}$ is emitted representing whether the task was successfully completed. For simplicity, we only consider finite horizon, episodic tasks. Given a previously collected dataset of videos labeled with task descriptions $\mathcal{D} = \{(\mathbf{x}, g)\}$, one can leverage behavioral cloning (BC) [162] to learn π by minimizing

$$\mathcal{L}_{\text{BC}}(\pi) = \mathbb{E}_{(\mathbf{x},g)\sim\mathcal{D}}[-\log \pi(\mathbf{x}|x_0, g)]. \tag{2.1}$$

Equation 2.1 can be viewed as maximizing the likelihood of the videos in $\mathcal{D}$ conditioned on the initial frame and task description.

2.3.2 Consistency Models

Diffusion models [87, 192] have emerged as an important technique for generative modeling of high-dimensional data. During training, a diffusion model learns to map noisy data (at various noise levels) back to clean data in a single step. Concretely, let $x^{(0)}$ denote a clean image and $x^{(t)}$ denote the noisy image at noise level t, where $t \in [0, T]$, the training objective for a diffusion model $f_\theta(x^{(t)}, t)$ can be written as

$$\mathcal{L}_{\text{diffusion}}(\theta) = \mathbb{E}_{x^{(0)},\epsilon,t}\left[\|f_\theta(x^{(t)}, t) - x^{(0)}\|^2\right], \tag{2.2}$$

where $\epsilon \in \mathcal{N}(0, I)$ is the added noise, and $x^{(t)} = \sqrt{\alpha_t}x^{(0)} + \sqrt{1-\alpha_t}\epsilon$ where α_t are time-dependent noise levels. Although diffusion models have achieved high-quality image/video generation, they require hundreds or thousands of denoising steps during inference, which induces tremendous computational cost. To overcome the slow sampling speed of diffusion

[1] We use $\Delta(\cdot)$ to denote a probability simplex function

models, *consistency models* [191, 190] were initially proposed by enforcing a consistency loss across different noise levels, i.e.,

$$\mathcal{L}_{\text{consistency}}(\theta) = \mathbb{E}_{x^{(0)},\epsilon,t_1,t_2}\left[\|f_\theta(x^{(t_1)},t_1) - \texttt{stopgrad}\big(f_\theta(x^{(t_2)},t_2)\big)\|^2\right], \tag{2.3}$$

which encourages the output of the single-step map between different noise levels to be similar. In fact, both the diffusion loss in Equation 2.2 and the consistency loss in Equation 2.3 can be understood as exploiting the structure of the denoising procedure which corresponds to an ordinary differential equation (ODE). Specifically, as introduced in [191, 189], the backward denoising procedure of a diffusion model can be characterized by an ODE, i.e.,

$$\frac{\mathrm{d}x^{(t)}}{\mathrm{d}t} = -t \cdot s(x^{(t)},t), \tag{2.4}$$

with $s(x^{(t)},t)$ is some score function. During the entire path along $t \in (\epsilon,\infty]$, following this ODE should always map $x^{(t)}$ to $x^{(0)}$. If we parametrize the model $f(x^{(t)},t)$ as the simulation following the ODE governed by $s(x^{(t)},t)$, we obtain the diffusion loss (2.2). Meanwhile, for all $t,t' \in (\epsilon,\infty]$, we have $f(x^{(t)},t) = f(x^{(t')},t')$ along the simulation path, which induces the consistency loss (2.3). Therefore, we can combine the diffusion loss and consistency loss together for model training, i.e.,

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{diffusion}}(\theta) + \lambda \cdot \mathcal{L}_{\text{consistency}}(\theta), \tag{2.5}$$

where λ denotes consistency regularization hyperparameter across different noise levels.

2.4 Video Generation as An Agent

In this section, we introduce VideoAgent , a framework for improving video plan generation. In Section 2.4.1, we develop *self-conditioning consistency* to iteratively refine generated video plans. In Section 2.4.2, we describe how a diffusion model trained with self-conditioning consistency can leverage inference-time compute to select the best video plans. Finally, in Section 2.4.3, we illustrate how VideoAgent closes the self-improvement loop by collecting additional online data to further enhance video generation and refinement.

2.4.1 Refinement through Self-Conditioning Consistency

We consider first-frame-and-language conditioned video generation following [50, 109], which generates a sequence of image frames to complete the task described by the language starting from the initial image. In practice, generated videos often contain hallucinations [238]. While such inaccuracies may prevent a video plan from fully completing the task,

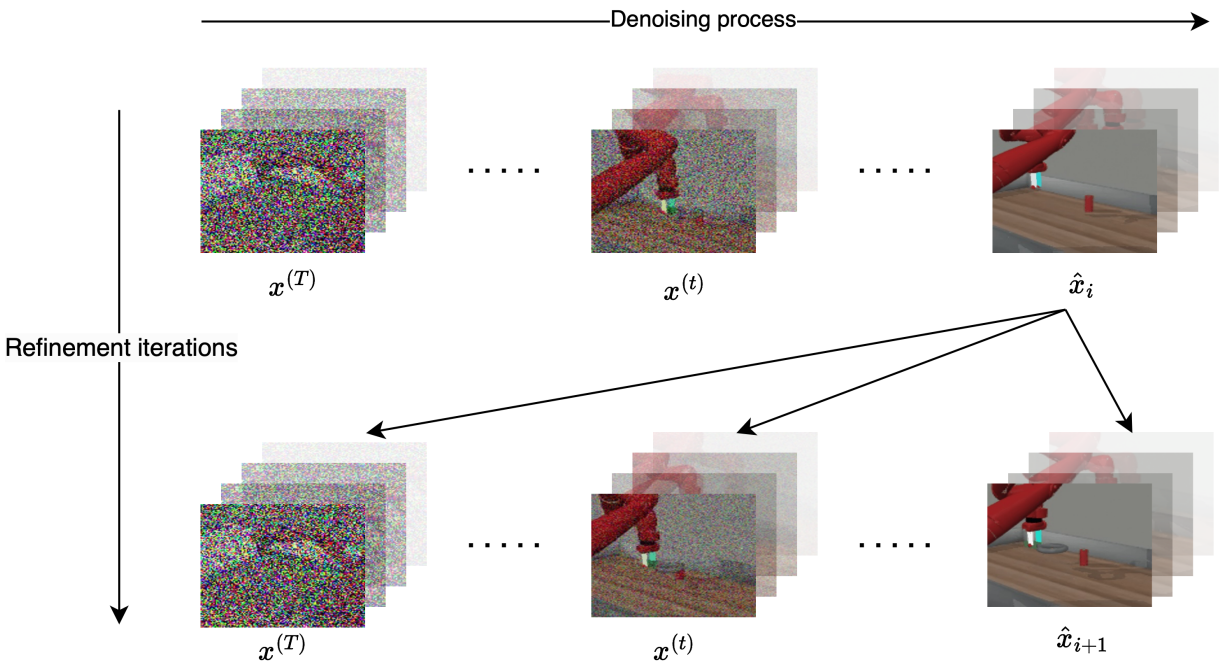


Figure 2.2: An illustration of Self-Conditioning Consistency. The horizontal direction represents regular denoising process. The two rows represent two refinement iterations. $\hat{\mathbf{x}}_i$ denotes generated video plan at refinement iteration i. We condition the refinement iteration $i+1$ on generated video from the previous iteration $\hat{\mathbf{x}}_i$.

the generated video may still make meaningful progress towards completing the task. Thus, instead of independently sampling many videos hoping that one may be free from hallucinations, we propose refining previously generated videos iteratively.

Specifically, let $\mathbf{x}^{(0)}$ denote the ground truth video and $\hat{\mathbf{x}}$ a generated video from the original T2V diffusion model. We introduce a *self-conditioning consistency* model, $\hat{f}_\theta(\hat{\mathbf{x}}, \mathbf{x}^{(t)}, t)$, which takes a generated video $\hat{\mathbf{x}}$ and a noisy version of the ground truth $\mathbf{x}^{(t)}$ as inputs to predict the clean video. This formulation enables iterative refinement by conditioning the model on its previous predictions, as illustrated in Figure 2.2. We denote video samples from the refinement model after the i-th iteration as $\hat{\mathbf{x}}_i$. Self-conditioning is inspired by a reparameterization of the implicit ODE solver for Equation 2.4 [189, 135, 254, 36]. For instance, [189] considered the first-order ODE solver for Equation 2.4 following:

$$\mathbf{x}^{(t-1)} = \sqrt{\alpha_{t-1}}\mathbf{x}^{(\mathbf{0})} + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot s(\mathbf{x}^{(t)}, t). \tag{2.6}$$

In VideoAgent , we adapt Equation 2.6 by replacing the $\mathbf{x}^{(0)}$ term with $\hat{\mathbf{x}}$, the previously generated video sample, as illustrated in Figure 2.2. In standard DDIM-based methods [189], $\mathbf{x}^{(0)}$ is typically obtained as an intermediate estimate from $\mathbf{x}^{(t)}$ within the *same* iteration. In contrast, our approach reuses $\hat{\mathbf{x}}$ from a *previous* iteration, allowing for a self-conditioning mechanism that improves temporal coherence. The self-conditioned consistency loss also ensures that compounding of errors do not propagate as it teaches the model to focus on the difference between the consecutive generations and regresses that to the ground truth video enabling the policy to learn to correct only the errors over each iteration. By enforcing consistency across iterations, our method thus enables the denoising process to correct potential failures more effectively.

We learn the ODE solver through self-conditioning consistency by directly predicting the clean video $\hat{\mathbf{x}}_{\mathbf{i+1}}$ using:

$$\begin{aligned}\mathcal{L}_{\text{SC-consistency}}(\theta) = \mathbb{E}_{\hat{\mathbf{x}},\mathbf{x}^{(0)},t}\Big[\|\hat{f}_\theta(\hat{\mathbf{x}}, \mathbf{x}^{(t)}, t) - \mathbf{x}^{(0)}\|^2\Big] \\ + \mu\mathbb{E}_{\hat{\mathbf{x}}_1,\hat{\mathbf{x}}_2,t}\Big[\|\hat{f}_\theta(\hat{\mathbf{x}}_1, \mathbf{x}^{(t)}, t) - \hat{f}_\theta(\hat{\mathbf{x}}_2, \mathbf{x}^{(t)}, t)\|^2\Big].\end{aligned} \tag{2.7}$$

The first term in Equation 2.7 represents the standard diffusion loss with the additional conditioning on $\hat{\mathbf{x}}$, while the second term regularizes the similarity between different refinement iterations ($\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$) to promote coherence across iterations. This iterative refinement process distinguishes self-conditioning consistency from traditional consistency models. Combined with the standard objective for video diffusion:

$$\mathcal{L}_{\text{video-diffusion}}(\theta) = \mathbb{E}_{\mathbf{x}^{(0)},\epsilon,t}\left[\|f_\theta(\mathbf{x}^{(t)}, t) - \mathbf{x}^{(0)})\|^2\right], \tag{2.8}$$

the overall objective for training a self-conditioning-consistent video diffusion model thus becomes:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{video-diffusion}}(\theta) + \lambda \mathcal{L}_{\text{SC-consistency}}(\theta). \tag{2.9}$$

Note that while the video generation model f_θ and the video refinement model $\hat{f}_\theta$ have different input arguments (first frame versus previously generated video), we can share their parameters to train a single unified model for both video generation and refinement tasks. This parameter-sharing approach allows us to leverage the same model architecture for generating initial video plans and iterative refinement. The training process for f_θ and $\hat{f}_\theta$ is detailed in Algorithm 1 in Section 2.5.

Feedback Guided Self-Conditioning Consistency. While we can refine videos only from previously generated samples, it may be desirable to condition the refinement process on any additional feedback for the previously generated video that is available (e.g., feedback from humans or vision language models critiquing which part of the generated video is unrealistic). Given we work with a goal-conditioned video policy setup, we have a lot of flexibility to adapt and use diverse feedback mechanisms - be it binary feedback from the VLM or from direct rollout in the environment, or detailed suggestive feedback from the VLM defining what went wrong and how to correct the plans. We can simply concatenate such feedback with the initial task goal and better guide the plan generation. When such feedback is available, we can have the refinement model $\hat{f}$ further take the additional feedback as input, combined with the task description, to guide the refinement process, i.e.,

$$\hat{f}_\theta(\mathbf{x}, \mathbf{x}^{(t)}, t|\text{feedback}), \tag{2.10}$$

which can be plugged into our framework for learning feedback-guided self-refinement using Equation 2.9.

2.4.2 Inference-Time Planning through VLM-Guided Video Generation.

After training the video generation model f_θ and the video refinement model $\hat{f}_\theta$ described in Equation 2.8 and Equation 2.7, we can sample from f_θ and iteratively apply $\hat{f}_\theta$ for video refinement. Specifically, let η be the step size for the noise schedule, σ_t be a time dependent noise term, VideoAgent first generates a video plan through

$$\mathbf{x}^{(t-1)} = \mathbf{x}^{(t)} - \eta \cdot \nabla_\theta f_\theta(\mathbf{x}^{(t)}, t) + \sigma_t \cdot \epsilon. \tag{2.11}$$

The sample $\hat{\mathbf{x}}$ after T denoising steps corresponds to the generated video. Next, we can iteratively apply $\hat{f}_\theta$ to refine the generated video sample

$$\hat{\mathbf{x}}_{i+1} = \hat{f}_\theta(\hat{\mathbf{x}}_i, \mathbf{x}^{(t)}, t), \tag{2.12}$$

where i denotes the video refinement iteration, with $\hat{\mathbf{x}}_0 = \hat{\mathbf{x}} = \mathbf{x}^{(T)}$. We denote the final video after refinement as $\hat{\mathbf{x}}_{\text{refined}}$. A natural question is when to stop the iterative video refinement process. We use a VLM as a proxy for the environment's reward to assess whether a refined video is likely to lead to successful execution in the environment. Specifically, we denote a VLM as $\hat{\mathcal{R}}$, which takes a refined video $\hat{\mathbf{x}}_i$ and returns a binary value $\{0, 1\}$ to determine whether a video is acceptable based on overall coherence, adherence to physical laws, and task completion (See prompt for VLM in Section 2.10). With $\hat{\mathcal{R}}$, the refinement stops when the VLM decides that the refined video is acceptable. Namely, we have

$$\hat{\mathbf{x}}_{\text{refined}} = \hat{\mathbf{x}}_{i^*}, \quad \text{where} \quad i^* = \min \left\{ i : \hat{\mathcal{R}}(\hat{\mathbf{x}}_i) = 1 \right\} \tag{2.13}$$

Algorithm 2 in Section 2.5 shows details for video plan generation, refinement, and selection during inference.

2.4.3 Self-Improvement through Online Finetuning

In addition to video refinement through self-conditioning consistency and inference-time compute as described in Section 2.4.1 and Section 2.4.2, we can further characterize the combination of video generation and video refinement as a policy, which can be improved by training on additional data collected from the environment during online interaction. Specifically, the goal is to maximize the expected returns of a policy through trial-and-error interaction with the environment:

$$\mathcal{J}_{\text{online}}(\theta) = \mathbb{E}\left[\mathcal{R}(\mathbf{a}) \,|\, \pi_\theta, \rho, \mathcal{E}\right], \tag{2.14}$$

where $\mathcal{R}$ is the true reward function, $\mathcal{E}$ is the interactive environment, and π_θ corresponds to the combination of f_θ and $\hat{f}_\theta$.

A broad array of reinforcement learning methods [201] such as policy gradient [177] can be employed to maximize the objective in Equation 2.14. For simplicity, we consider the setup of first executing the policy in the environment, then filtering for successful trajectories, continuing finetuning the video policy using additional online data, and executing the finetuned policy again to collect more data. Specifically, each online iteration constructs an additional dataset by rolling out the policy π_θ at the current online iteration

$$\mathcal{D}_{\text{new}} = \left\{ \hat{\mathbf{x}}_{\text{refined}} \sim \pi_\theta(x_0, g) \mid \mathcal{R}(\rho(\hat{\mathbf{x}}_{\text{refined}})) = 1 \right\}, \tag{2.15}$$

where ρ is the optical flow model that maps the refined video to low-level control actions. See Algorithm 3 in Section 2.5 for details of online policy finetuning.

2.4.4 Unsupervised Policy Learning via Implicit Successor Measures

The VideoAgent framework, while presented as a generative approach to embodied planning, can be effectively contextualized within the paradigm of unsupervised, reward-free reinforcement learning (RL) [250]. Specifically, its core mechanisms for goal-conditioned video generation and self-refinement draw strong parallels to the concept of **successor measures**, a representation that enables efficient, zero-shot adaptation to new tasks by decoupling environment dynamics from task-specific rewards [23, 266, 171]. This perspective clarifies VideoAgent's contribution as a novel, high-dimensional implementation of these advanced RL principles.

The conventional approach in reward-free RL involves a two-stage process: (1) an unsupervised pre-training phase where an agent learns the environment's dynamics, and (2) a test phase where a new reward function is provided, and the agent must generate a near-optimal policy without further environment interaction. A key enabler for this is the successor measure (SM). For a policy π, the successor measure $M^\pi(x_0, \cdot)$ represents the discounted cumulative distribution of future states visited after starting in state x_0. Mathematically, it is defined as:

$$M^\pi(x_0, X) := \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(x_t \in X | x_0, \pi) \quad \forall X \subseteq \mathcal{X} \tag{2.16}$$

By learning this predictive map, an agent can rapidly compute the value function for any new reward function r_g (specified by goal g) by integrating the reward over the successor measure:

$$V^\pi_{r_g}(x_0) = \int_{x'} r_g(x') dM^\pi(x_0, x') \tag{2.17}$$

VideoAgent's methodology implements this paradigm through its generative components, creating a clear mathematical analogy at each stage:

1. **Unsupervised Pre-training as Dynamics Learning:** The initial training of the video generation policy π is analogous to the unsupervised, reward-free pre-training phase. In traditional RL, this phase aims to learn the transition dynamics $P(x_{t+1}|x_t, a_t)$. VideoAgent achieves a similar goal by learning a generative policy that models entire trajectories. The behavioral cloning objective from **Equation 2.1** trains the policy to maximize the likelihood of observed video data:

$$\mathcal{L}_{BC}(\pi) = \mathbb{E}_{(x,g)\sim\mathcal{D}}[-log\ \pi(x|x_0, g)] \tag{2.18}$$

This objective directly trains π to be a generative model of the environment's dynamics as represented in the dataset $\mathcal{D}$. The policy learns to produce sequences of states $[x_0, ..., x_H]$ that are plausible under the true dynamics, effectively creating a generator for the successor measure distribution.

2. **Video Plans as Implicit, High-Dimensional Successor Measures:** A generated video plan, $x = [x_0, ..., x_H]$, produced by the policy $\pi(\cdot|x_0, g)$, is a high-dimensional, non-parametric sample from the distribution implicitly defined by the successor measure M^π. While a traditional SM is a function that provides an *expected* future state occupancy, VideoAgent's generative policy provides a concrete *sample* of a full future trajectory. This is a richer, more expressive representation than a low-dimensional successor feature vector.

3. **VLM-Guidance as Zero-Shot Task Specification and Planning:** The inference-time refinement process corresponds to the test phase in zero-shot RL, where a policy is derived for a new reward function. The VLM, denoted as $\hat{\mathcal{R}}$, acts as a proxy for a zero-shot reward function $r_g(\cdot)$. In the SM framework, the goal is to find a policy that maximizes the value $V^\pi_{r_g}(x_0)$. VideoAgent's refinement and selection process is a practical implementation of this optimization. The iterative refinement in **Equation 2.12**, $\hat{x}_{i+1} = \hat{f}_\theta(\hat{x}_i, x^{(t)}, t)$, performs a search in the space of video plans to find a trajectory that maximizes the implicit reward from the VLM. The final selection in **Equation 2.13** identifies the plan that satisfies the VLM's criteria:

$$\hat{x}_{refined} = \hat{x}_{i^*}, \text{ where } i^* = min\{i : \hat{\mathcal{R}}(\hat{x}_i) = 1\} \tag{2.19}$$

This is analogous to finding a trajectory sample $\hat{x}_{refined}$ from the successor measure that yields a high value, effectively maximizing the task-specific value function at inference time.

4. **Self-Improvement as Policy Refinement:** The online finetuning stage refines the underlying generative policy π_θ to better align with true task success. In traditional RL, policy improvement adjusts policy parameters θ to increase the expected return, often via policy gradients: $\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}$. VideoAgent's online objective in **Equation 2.14** similarly aims to maximize the expected true environment reward $\mathcal{R}$:

$$\mathcal{J}_{online}(\theta) = \mathbb{E} \tag{2.20}$$

This is achieved by collecting a new dataset $\mathcal{D}_{new}$ of successful trajectories (**Equation 2.15**) and using it to update the model parameters θ by minimizing the combined loss from **Equation 2.9**. While this is a form of behavioral cloning on successful

data rather than a direct policy gradient update, the effect is analogous: it shifts the probability mass of the policy π_θ towards generating trajectories that yield high rewards. This improves the model's ability to generate more task-successful successor measures, enhancing its internal world model based on real-world feedback.

We thus can view VideoAgent is not just from the lens of a generative simulator or world model but also as an instantiation of a goal-conditioned, unsupervised agent policy. It learns an implicit, high-dimensional successor measure during a reward-free pre-training phase and leverages goal-conditioning from diverse VLM or environmental feedback as a zero-shot reward function for inference-time planning and refinement.

2.5 Algorithms

Algorithm 1: Training of Video Generation and Refinement Models with VLM Feedback

Input: Dataset $\mathcal{D}$, learning rate γ, total training iterations N, initial model parameters θ, video generation model f_θ, video refinement model $\hat{f}_\theta$, VLM $\hat{\mathcal{R}}$

for *iteration* $= 1$ **to** N **do**

Sample $\{(\mathbf{x}^{(0)}, g)\} \sim \mathcal{D}$ and $t \sim \text{Uniform}(\{0, 1, \ldots, T\})$;

Compute vanilla diffusion loss:

$$\mathcal{L}_{\text{video-diffusion}} = \left\| f_\theta(\mathbf{x}^{(t)}, t) - \mathbf{x}^{(0)} \right\|^2;$$

Generate $\hat{\mathbf{x}}$ following Equation 2.11 and sample $\texttt{feedback} \sim \hat{\mathcal{R}}(\cdot|\hat{\mathbf{x}})$;

Compute consistency loss:

$$\mathcal{L}_{\text{SC-consistency}} = \left\| \hat{f}_\theta(\hat{\mathbf{x}}, \mathbf{x}^{(t)}, t \,|\, \texttt{feedback}) - \mathbf{x}^{(0)} \right\|^2;$$

Update parameters:

$$\theta \leftarrow \theta - \gamma \nabla_\theta \left(\mathcal{L}_{\text{video-diffusion}} + \mathcal{L}_{\text{SC-consistency}} \right);$$

Algorithm 2: VLM Guided Replan

Input: Initial frame x_0, task description g, Reward $\mathcal{R}$, Environment $\mathcal{E}$, VLM $\hat{\mathcal{R}}$, max_refine_iterations, max_replans

for *replan_count* $= 1$ **to** *max_replans* **do**
- $\hat{\mathbf{x}} \leftarrow \pi_\theta(x_0, g)$;
- **for** $i = 0$ **to** *max_refine_iterations* **do**
  - response $\leftarrow \hat{\mathcal{R}}(\hat{\mathbf{x}}_{(i)}, g)$;
  - **if** response == ACCEPT then **break**;
  - $\hat{\mathbf{x}}_{(i+1)} \leftarrow \pi_\theta(\hat{\mathbf{x}}_{(i)}, x_0, g)$;
- success $\leftarrow \mathcal{R}(\rho(\hat{\mathbf{x}}_{\text{refined}}))$;
- **if** success then **break**;
- $x_0 \leftarrow \mathcal{E}$.get_state();

Algorithm 3: Online Finetuning of Video Generation and Refinement Models

Input: Dataset $\mathcal{D}$, policy π_θ, Reward $\mathcal{R}$, Environment $\mathcal{E}$

for *iteration* $i = 1$ *to* N **do**
- $\mathcal{D}_{\text{new}} \leftarrow \emptyset$;
- **for** *each* $(\cdot, g)$ *in* $\mathcal{D}$ **do**
  - $x_0 \leftarrow \mathcal{E}$.reset$(g)$;
  - $\hat{x}_{\text{refined}} \sim \pi_\theta(x_0, g)$;
  - **if** $\mathcal{R}(\rho(\hat{x}_{refined}))$ **then**
    - $\mathcal{D}_{\text{new}} \leftarrow \mathcal{D}_{\text{new}} \cup (\hat{x}_{\text{refined}}, g)$;
- $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_{\text{new}}$;
- Finetune θ using Algorithm 1 on $\mathcal{D}$;

2.6 Dataset Descriptions in Detail

2.6.1 Datasets and Environments.

We follow the same evaluation setting as [109], which considers three datasets: Meta-World [245], iTHOR [113], and BridgeData V2 [210]. Meta-World consists of 11 robotic manipulation tasks performed by a simulated Sawyer arm, with video demonstrations captured from three distinct camera angles. iTHOR is a simulated 2D object navigation benchmark, where an agent searches for specified objects across four room types. BridgeData V2 is a real-world dataset of robotic manipulation.

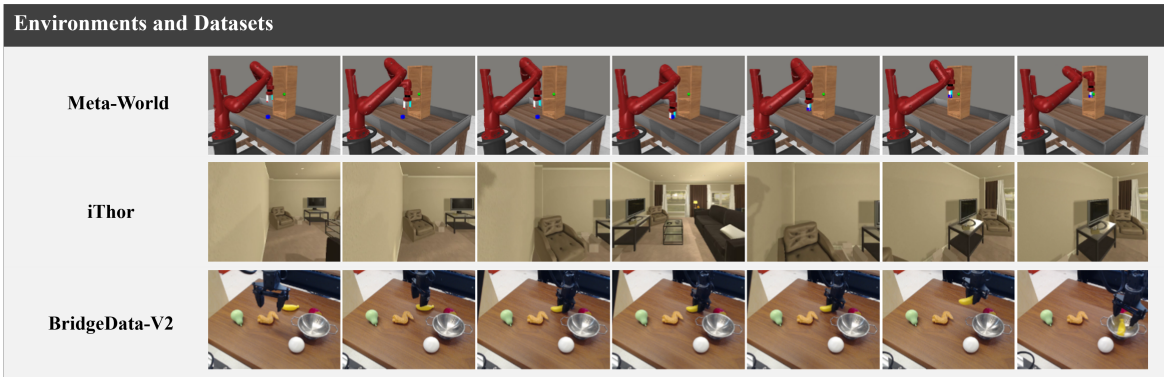


Figure 2.3: Environments and Datasets that we work with: Meta-World, iThor, and BridgeData-V2

Meta-World [245] is a simulation benchmark that uses a Swayer robotic arm to perform a number of manipulation tasks. In our experiments, we make use of 11 tasks as shown in Table 2.1. We capture videos from three distinct camera angles for each task and use the same camera angles for both the training and testing phases. We gather five demonstration videos per task for each camera angle. During the evaluation, we tested on each of the three camera angles with 25 seeds per camera angle. The position of the robot arm and the object is randomized at the beginning of each seed to ensure variability. A trajectory is considered successful if the Video Agent reaches within a really close threshold of the goal state.

iTHOR [113] is another popular 2D simulated benchmark that focuses on embodied common sense reasoning. We evaluate the Video as Agent framework on the object navigation tasks, where an agent is randomly initialized in a scene and tasked with finding an object of a specified type (e.g., toaster, television). At each time step, the agent can take one of the four possible actions (MoveForward, RotateLeft, RotateRight, or Done), and observes a 2D scene to operate in. We selected 12 objects ((e.g. toaster, television) to be placed in 4 different room types (e.g. kitchen, living room, bedroom, and bathroom). Again, the starting position of the agent is randomized at the start of each episode. During evaluation, we test the agent across 12 object navigation tasks spread across all 4 room types, 3 tasks per room. A trajectory is successful if the agent views and reaches within 1.5 meters of the target object before reaching the maximum environment step or predicting Done.

To test the usefulness of our framework across different videos types, we also use the BridgeData V2 dataset [210], a large and diverse dataset of real world robotic manipulation behaviors designed to facilitate research in scalable robot learning. It contains 60,096

trajectories collected across 24 environments using a publicly available low-cost WidowX 250 6DOF robot arm. The dataset provides extensive task and environment variability, enabling skills learned from the data to generalize across environments and domains.

2.6.2 Additional trajectories per iteration during online training

We collect 15 successful trajectories for each task during every iteration. This standardization helps address task imbalance, as task success rates are higher for certain tasks compared to others. By ensuring a fixed number of successful trajectories per task, we prevent overfitting to easier tasks and maintain balanced model performance across the entire task set. The set of seeds used for training and collecting additional trajectories are different from the seeds used for evaluation.

2.7 Experiments

In this section, we evaluate the performance of VideoAgent . First, we measure the end-to-end task success rate of VideoAgent against the baselines in Section 2.7.2, and study the effect of different components of VideoAgent in Section 2.7.3. Finally, we show that VideoAgent is effective in improving the quality of real robotic videos in Section 2.7.5.

Evaluation Setup. We follow the evaluation setup of AVDC [109], a method for controlling simulated robots using dense action correspondence extracted from generated videos. We follow AVDC and perform evaluation in three environments: Meta-World [245], iTHOR [113], and BridgeData V2 [210] (see detailed dataset description in Section 2.6). We compare variants of VideoAgent , including VideoAgent with only self-refinement (Section 2.4.1), VideoAgent -Online (Section 2.4.3), and VideoAgent -Replan against AVDC and AVDC-Replan (replanning when movement stalls) from [109]. More detailed descriptions of the baselines and the VideoAgent variants are provided below.

2.7.1 Baselines and VideoAgent Variants.

We consider the following methods for comparison:

- **AVDC** (baseline). This is the Actions from Video Dense Correspondences [109] baseline, which synthesizes a video and predicts optical flow to infer actions.
- **AVDC-Replan** (baseline). When the movement stalls, AVDC-replan re-runs video generation and action extraction from the flow model to execute a new plan.
- **VideoAgent** . Our proposed video refinement model through self-conditioning consistency as introduced in Section 2.4.1. VideoAgent generates video and iteratively refines

a video plan. We use GPT-4 Turbo for selecting the best video plan during inference (Section 2.4.2).

- **VideoAgent -Online**. As actions are executed in the online environment, successful trajectories are collected and used to continue training the video generation and refinement model, as described in Section 2.4.3.
- **VideoAgent -Replan**. This variant incorporates online filtering of successful trajectories with the replanning mechanism, where replanning is conducted first, and more successful trajectories after replanning are added back to the training data.

2.7.2 End-to-End Task Success

Meta-World. We report the task success rates of baselines and VideoAgent in Table 2.1. Following [109], we evaluate performance across three camera poses with 25 seeds per pose. Without online environment access, VideoAgent improves the overall success rate through self-conditioning consistency alone from 19.6% (AVDC) to 22.3%. For certain difficult tasks, e.g., faucet-close, VideoAgent improves performance from 12.0% to 46.7%. With online data collection, VideoAgent -Online further improves success rates with each additional online iteration of rolling out the policy, collecting successful trajectories, and finetuning. VideoAgent -Online can be further combined with replanning, achieving 53.7% overall success, surpassing prior state-of-the-art on this benchmark. Detailed baseline comparisons are provided in Section 2.8.2, and qualitative improvements in refined videos are shown in Figure 2.9 in Section 2.12.

iTHOR. Next, we evaluate VideoAgent on iThor. Due to the high computational cost of running the iThor simulator(approx. 20 minutes per roll-out), we focus only on evaluating self-conditioning consistency (without online access). We follow the same setup as [109], where we measure the average success rate across four rooms each with three objects using 20 seeds. As shown in Table 2.2, VideoAgent consistently outperforms the AVDC baseline, demonstrating the effectiveness of self-conditioning consistency in producing more plausible video plans for first-person view navigation (i.e., what iThor measures).

2.7.3 Effect of Different Components in VideoAgent

In this section, we aim to understand the effect of different components of VideoAgent . Specifically, we focus on the effect of (1) different types of feedback given to the refinement model, (2) the number of refinement and online iterations, and (3) the quality of the VLM feedback.

	door-open	door-close	basketball
AVDC	30.7%	28.0%	21.3%
AVDC-Replan	72.0%	89.3%	37.3%
VideoAgent	40.0%	29.3%	13.3%
VideoAgent-Online (Iter1)	48.0%	40.0%	24.0%
VideoAgent-Online (Iter2)	58.7%	50.7%	28.0%
VideoAgent-Online-Replan	**82.7%**	**97.3%**	**40.0%**

	shelf-place	btn-press	btn-press-top
AVDC	8.0%	34.7%	17.3%
AVDC-Replan	18.7%	60.0%	24.0%
VideoAgent	9.3%	38.7%	18.7%
VideoAgent-Online (Iter1)	12.0%	42.7%	36.0%
VideoAgent-Online (Iter2)	18.7%	53.3%	41.3%
VideoAgent-Online-Replan	**26.7%**	**73.3%**	**44.0%**

	faucet-close	faucet-open	handle-press
AVDC	12.0%	17.3%	41.3%
AVDC-Replan	53.3%	24.0%	81.3%
VideoAgent	46.7%	12.0%	36.0%
VideoAgent-Online (Iter1)	53.3%	28.0%	52.0%
VideoAgent-Online (Iter2)	58.7%	36.0%	64.0%
VideoAgent-Online-Replan	**74.7%**	**46.7%**	**86.7%**

	hammer	assembly	**Overall**
AVDC	0.0%	5.3%	19.6%
AVDC-Replan	8.0%	6.7%	43.1%
VideoAgent	0.0%	1.3%	22.3%
VideoAgent-Online (Iter1)	1.3%	5.3%	31.2%
VideoAgent-Online (Iter2)	1.3%	9.3%	38.2%
VideoAgent-Online-Replan	**8.0%**	**10.7%**	**53.7%**

Table 2.1: Meta-World Results. Mean success rates of baselines and VideoAgent on 11 simulated robot manipulation environments from Meta-World. VideoAgent consistently outperforms baselines across all tasks.

Room	AVDC	VideoAgent
Kitchen	26.7%	**28.3%**
Living Room	23.3%	**26.7%**
Bedroom	38.3%	**41.7%**
Bathroom	36.7%	**40.0%**
Overall	31.3%	**34.2%**

Table 2.2: iThor Success Rates comparing VideoAgent with the AVDC baseline. VideoAgent outperforms AVDC across all four rooms averaged over all three objects in each room.

Effect of Different VLM Feedback

In the previous section, we only used VLM during inference to determine when to stop refining a generated video. However, it is natural to wonder if information-rich feedback from the VLM, such as language descriptions of which part of a generated video to improve, might lead to better refined videos. To answer this question, we propose a few variants of VideoAgent according to the feedback available when training the video refinement model as in Equation 2.10. Specifically, we use VideoAgent to denote training the video refinement model only conditioned on the original task description. VideoAgent -Binary denotes additionally conditioning on whether a generated video is determined to be successful by the VLM. VideoAgent -Suggestive denotes conditioning additionally on language feedback from the VLM on which part of the video needs improvement and how the video can be improved. We train these three versions of the video refinement model, and report the overall task success from Meta-World in Table 2.3. We see that VideoAgent-Binary improves upon the base VideoAgent, while training with descriptive feedback in VideoAgent-Suggestive leads to even better performance. This suggests that richer feedback from the VLM can facilitate better training of the video refinement model. Improvement for each individual task can be found in the Section 2.7.4.

	Overall
AVDC	19.6%
VideoAgent	22.3%
VideoAgent-Binary	23.8%
VideoAgent-Suggestive	26.6%

Table 2.3: Effect of Different Feedback used to train the refinement model. Descriptive feedback from the VLM leads to higher improvement in task success.

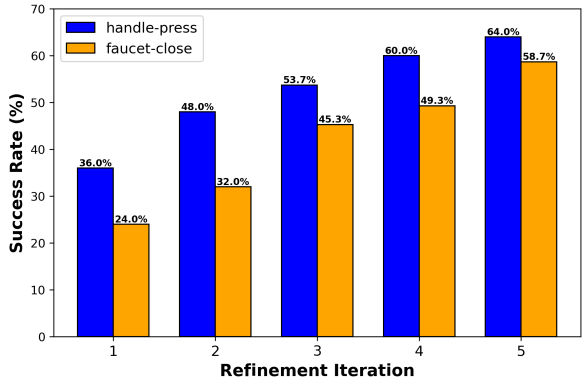


Figure 2.4: Effect of Refinement Iterations. The accuracy of downstream tasks generally increases as the number of refinement iteration increases.

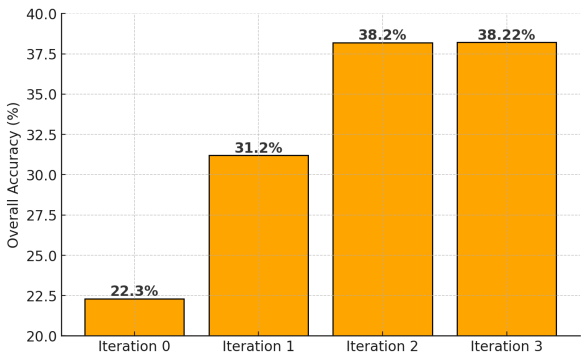


Figure 2.5: Effect of Online Iterations. The overall task success of VideoAgent increases as the number of online iterations increases.

Effect of the Number of Iterations.

Next, we want to understand whether more refinement iterations and online finetuning iterations lead to higher task success. We found that while different tasks require a different number of refinement and online iterations to achieve the best performance, VideoAgent does perform better as the number of refinement and online iterations increases, as shown in Figure 2.4 and Figure 2.5. During video refinement, specific tasks such as handle-press and faucet-close continue to show improvement even at the fifth refinement iteration. Faucet-close especially benefits from more refinement iterations, bringing success rate from 24.0% to 58.7% after five refinement iterations. The improved task success rates across refinement and online iterations suggests that self-conditioning consistency discussed in Section 2.4.1 and online interaction discussed in Section 2.4.3 can indeed effectively reduce hallucination and improve physical plausibility in the generated videos.

Accuracy of VLM feedback

	Precision	Recall	F1-Score	Accuracy
Unweighted	0.65	0.89	0.76	0.69
Weighted	**0.92**	0.58	0.71	0.75
Without Cam 3	0.91	0.71	0.80	0.79

Table 2.4: VLM Performance measured according to whether a VLM considers a generated video as acceptable using human label as the ground truth.

Since VideoAgent heavily relies on a VLM to select video plans during inference, it is crucial to understand whether the VLM can in fact achieve a reasonable accuracy in providing feedback for video generation. To quantify the performance of a VLM, we use human labels on whether a generated video is acceptable as the ground truth, and measure precision, recall, F1-score, and accuracy based on whether GPT-4 Turbo thinks the generated video is acceptable according to trajectory smoothness (consistent across sequential frames), physical stability, and achieving the goal (See full prompt in Section 2.10). We report the average result across 36 generated videos from the Meta-World dataset in Table 2.4. We see that the original prompt we used (Unweighted, meaning not emphasizing reduction of false positives) achieves 69% accuracy, suggesting that the VLM is able to somewhat judge generated videos but not always accurately. Since VideoAgent uses multiple refinement iterations, we want to avoid false positives where a bad video is accidentally accepted. We can achieve this by penalizing false positives through reweighing its cost in the prompt, which leads to the VLM rejecting videos when the VLM is uncertain about the video's acceptability. This adjustment results in a significant increase in precision as shown in Table 2.4. This weighted version of the prompt is used in the experiments in Section 2.7.2.

Partial Observability. In the AVDC experimental setup, center cropping the third camera (what is used in the pipeline) often results in most of the robot arm being outside of the frame. We found that the accuracy of the VLM is affected by such partial observability. As shown in Table 2.4, removing the third camera from the prompt leads to much higher accuracy.

Descriptive Feedback. While VLM can provide binary feedback on whether a generated video is acceptable, we also measure the accuracy of the VLM in giving more descriptive feedback such as identifying the issue and providing suggestions on how to improve the video. We use three examples with human written language feedback as prompt for in-context learning. GPT-4 Turbo achieves 73.5% accuracy on identification and 86.1% accuracy on suggestion, as evaluated by humans. This result is highly encouraging and opens up future directions of leveraging descriptive feedback from VLMs to improve video generation.

2.7.4 VLM Feedback for Correction

2.7.5 Evaluating Self-Refinement on Real-World Robot Videos

In this section, we evaluate VideoAgent 's ability to refine real-world videos, which often contain higher variability, intricate details, nuanced behaviors, and complex interactions. We study the effect of video refinement using both quantitative metrics and qualitatively for holistic evaluation.

We report the average across 2250 videos generated from the AVDC baseline and from VideoAgent in Table 2.6. VideoAgent performs better according to all metrics except for Dynamic Degree from Video-Score (which shows similar performance between VideoAgent and AVDC). Notably, the gain is significant for metrics critical for instruction following and real-world videos, such as CLIP Score, Factual Consistency, and Text-to-Video Alignment. Improvement in Flow Consistency and Temporal Consistency suggests that VideoAgent produces smoother and more physically plausible videos that adhere better to the physical constraints of the real-world. This directly translates to better performance in real-world robotic tasks in Table 2.1.

Qualitative Evaluation. Next, we qualitatively evaluate generated videos from the AVDC baseline and from VideoAgent . We collect 50 generated videos from each model and conduct qualitative evaluation on whether a generated video looks realistic. Videos with refinement from VideoAgent improves the acceptance rate by 22% as shown in Table 2.6. We further show an example video with and without refinement in Figure 2.8 which we provide in Section 2.12, where the baseline (middle row) hallucinates (the bowl disappears) whereas VideoAgent produces the video that completes the task (bottom row). We also present a more fine-grained analysis of Visual Quality, Temporal Consistency, Dynamic Degree, Text to Video Alignment, and Factual Consistency evaluated qualitatively in the Section 2.7.6 with the metrics in Table 2.7, which further echos the results of qualitative evaluations presented in Table 2.6.

Quantitative Evaluation. Following previous literature on video generation, we consider two reference-free metrics, CLIP Score [82] and Flow Consistency [205], as well as a set of Video-Scores [77] for evaluation. CLIP Score measures the cosine similarity between frame feature and text prompt, whereas Flow Consistency measure the smoothness and coherence of motion in the videos calculated from the RAFT model. Video-Scores use five sub-metrics with a focus on correlation with qualitative evaluation and real-world videos.

2.7.6 Details of Qualitative Evaluation on BridgeData V2

Qualitative Evaluation. Next, we qualitatively evaluate video generation quality using the five Video-Score dimensions: Visual Quality (VQ) for clarity and resolution, Temporal Consistency (TC) for smooth frame transitions, Dynamic Degree (DD) for capturing accurate object/environment changes, Text-to-Video Alignment (TVA) for matching the video to the prompt, and Factual Consistency (FC) for adherence to physical laws and real-world facts. Videos are rated on a 4-point scale based on the metric in [77]: 1 (Bad), 2 (Average), 3 (Good), and 4 (Perfect). Our evaluation is based on 50 generated videos from a held-out set.

In terms of VQ and TC, both the baseline AVDC and our VideoAgent generate average quality videos (graded 2), with AVDC hallucinating more and generating some choppy jumps in videos temporally (we grade such videos as 1) and Video Agent fixing some of these upon video conditioned iterative refinement. The reason for AVDC baseline having higher DD is attributed to unruly movements that cause higher DD scores compared to VideoAgent, where movements are smoother. This also explains the result in fifth row of Table 2.6, and upon closer examination of the generated videos and their corresponding individual scores, we observed similar traits in videos having higher DD due to unnatural robot arm movements and object impermanence. TVA shows trends similar to ClipScore in Table 2.6 due to the better instruction following ability of VideoAgent leading to more controlled generation. FC is a very crucial metric for deployment of video generation agents as policy for task completion in robotics, scene navigation, and so on. Improved visual quality does not imply adherence to correct physical laws and real-world constraints, FC particularly checks for this aspect and due to video conditioned self-refinement, VideoAgent has better FC compared to AVDC.

Unsupervised policy learning methods like VideoAgent provide some great practical significance - because they achieve policy improvement without action labels or explicit rewards. By reducing hallucinations and producing physically plausible video plans, VideoAgent enables more reliable control when executed. Its inference-time refinement demonstrates that agents can improve their plans on the fly, making them more adaptable in novel scenarios. The framework illustrates how generative simulators, guided by diverse feedback, can serve as scalable tools for embodied planning.

2.8 Extended Experiments

2.8.1 Videos to action conversion

We employ the GMFlow optical flow model [232] to predict dense pixel movements across frames. These predicted flows serve as the foundation for reconstructing both object movements and robot motions depicted in the video. The flow predictions allow us to interpret the temporal evolution of the video in terms of actionable physical dynamics. The optical flow essentially provides a dense correspondence of pixel movements between consecutive frames, which is then used to infer the relative motion of objects and the robot. This mapping bridges the gap between the high-dimensional video representation and the low-level control commands required to execute the tasks in a simulated or real environment.

This method ensures that the generated video plans are actionable and aligned with the

task-specific dynamics, making the video generation process directly relevant to downstream policy learning and execution.

2.8.2 Baseline experiments on Metaworld

We conduct experiments on additional baselines including, Behavioral Cloning (BC), UniPi (with replan), VLP, AVDC-IS-Replan and Diffusion policy. Table 2.8 consists of these results. **VLP** follows a training setup similar to ours, but does not incorporate the proposed self-consistency loss. **AVDC-IS-Replan** refers to the baseline AVDC model with replanning and a straightforward inference-time scaling strategy, wherein the number of denoising time-steps is increased from 100 to 500 during inference. Our method surpasses all the above baselines considered.

2.8.3 Further Analysis of VideoAgent-Online

We train VideoAgent-Online for multiple iterations and observe that after 2 iterations, the results start to stabilize. The extra results for iteration 3 are also shown in table 2.9.

2.9 Architectural Details of VideoAgent

2.9.1 Video Diffusion training details

We use the same video diffusion architecture as the AVDC baseline. For all models, we use dropout=0, num head channels=32, train/inference timesteps=100, training objective=predict v, beta schedule=cosine, loss function=l2, min snr gamma=5, learning rate=1e-4, ema update steps=10, ema decay=0.999. All of our models and experiments were run on Nvidia A6000 GPUs.

2.9.2 Inference time speed

In our current setup, during inference, our video generation model produces a new video within 10 seconds on a single A6000 GPU at a resolution of 128×128 for Meta-World. The process of mapping this generated video to an action takes, on average, an additional 25 seconds. This action-mapping stage involves calculating optical flow, receiving feedback from the vision-language model (VLM), and converting the video into an action sequence based on the computed flow.

2.10 Prompt Structure for VLM Feedback

2.10.1 Binary Classification

We employ a structured prompting strategy to provide feedback on video sequences for the zero-shot classification. The process consists of one Query-Evaluation Phase, each with distinct sub-goals.

BINARY CLASSIFICATION

Task: You are a video reviewer evaluating a sequence of actions presented as seven consecutive image uploads, which together represent a single video. You are going to accept the video if it completes the task and the video is consistent without glitches.
Query-Evaluation Phase:

- **Inputs Provided:**
  - **Textual Prompt:** Describes the task the video should accomplish.
  - **Conditioning Image:** Sets the fixed aspects of the scene.
  - **Sequence of Images (7 Frames):** Represents consecutive moments in the video to be evaluated.
- **Evaluation Process:**
  - **View and Analyze Each Frame:** Examine each image in sequence to understand the progression and continuity of actions.
  - **Assess Overall Coherence:** Determine if actions transition smoothly and logically from one image to the next.
  - **Check for Physical Accuracy:** Ensure adherence to the laws of physics, identifying any discrepancies.
  - **Verify Task Completion:** Confirm the sequence accomplishes the task described in the textual prompt.
  - **Identify Inconsistencies:** Detect inconsistencies in object movement or overlaps that do not match the conditioning image.

BINARY CLASSIFICATION (Continued)

Task: You are a video reviewer evaluating a sequence of actions presented as seven consecutive image uploads, which together represent a single video. You are going to accept the video if it completes the task and the video is consistent without glitches.
Query-Evaluation Phase:

- **Evaluation Criteria:**
  - Accept the sequence if it is a coherent video that completes the task.
  - Reject the sequence if any frame fails to meet the criteria, showing inconsistencies or not achieving the task. Be very strict, rejecting even minor errors.
- **Response Requirement:**
  - Provide a single-word answer: *Accept* or *Reject*. Do not give reasoning.
- **Additional Notes:**
  - No further clarification can be requested.
  - Elements from the conditioning image must match those in each frame of the sequence.

2.10.2 Identification and Suggestion:

We employ a structured prompting strategy to provide descriptive feedback on video sequences via an in-context few-shot classification setup. The process consists of one Query-Evaluation Phase, each with distinct sub-goals.

Table 2.5: Meta-World: VideoAgent-Feedback Guided Results The mean success rates for various tasks, comparing different VideoAgent-Feedback Guided variants and the AVDC baseline.

	door-open	door-close	basketball
AVDC	30.7%	28.0%	21.3%
VideoAgent	40.0%	29.3%	13.3%
VideoAgent-Binary	46.7%	32.0%	14.7%
VideoAgent-Suggestive	46.7%	33.3%	18.7%
VideoAgent-Online-Suggestive	52.0%	28.0%	21.3%

	shelf-place	btn-press	btn-press-top
AVDC	8.00%	34.7%	17.3%
VideoAgent	9.3%	38.7%	18.7%
VideoAgent-Binary	6.7%	38.7%	21.3%
VideoAgent-Suggestive	12.0%	41.3%	22.7%
VideoAgent-Online-Suggestive	16.0%	46.7%	22.7%

	faucet-close	faucet-open	handle-press
AVDC	12.0%	17.3%	41.3%
VideoAgent	46.7%	12.0%	36.0%
VideoAgent-Binary	46.7%	17.3%	32%
VideoAgent-Suggestive	48.7%	17.3%	46.7%
VideoAgent-Online-Suggestive	45.3%	20.0%	48.0%

	hammer	assembly	**Overall**
AVDC	0.00%	5.30%	19.6%
VideoAgent	0.00%	1.3%	22.3%
VideoAgent-Binary	0.00%	5.3%	23.8%
VideoAgent-Suggestive	0.00%	5.3%	26.6%
VideoAgent-Online-Suggestive	2.7%	5.3%	27.4%

Metrics	AVDC	Video Agent
Clip Score	22.39	**22.90**
Flow Consistency	2.48 ± 0.00	**2.59 ± 0.01**
Visual Quality	1.97 ± 0.003	**2.01 ± 0.003**
Temporal Consistency	1.48 ± 0.01	**1.55 ± 0.01**
Dynamic Degree	**3.08 ± 0.01**	3.07 ± 0.02
Text to Video Alignment	2.26 ± 0.003	**2.30 ± 0.03**
Factual Consistency	2.02 ± 0.004	**2.07 ± 0.01**
Average Video Score	2.16 ± 0.01	**2.20 ± 0.01**
Qualitative Eval on Task Success	42.0%	**64.0%**

Table 2.6: BridgeData-V2 Results. Quantitative metrics comparing AVDC and VideoAgent on generated Bridge data. VideoAgent outperforms the baseline according to all except for one metric.

Metrics		AVDC	Video Agent
Task Success via Qualitative Eval		42.0%	**64.0%**
Holistic Assessment	Visual Quality	1.74	1.84
	Temporal Consistency	1.58	1.76
	Dynamic Degree	3.14	2.98
	Text to Video Alignment	2.66	3.04
	Factual Consistency	3.22	3.30
	Qualitative Eval Average	2.47	**2.98**

Table 2.7: Task Success and Other Fine-grained Qualitative Evaluation Metrics on BridgeData-V2

	door-open	door-close	basketball	shelf-place	btn-press	btn-press-top
BC-Scratch	21.3%	36.0%	0.0%	0.0%	34.7%	12.0%
BC-R3M	1.3%	58.7%	0.0%	0.0%	36.0%	4.0%
UniPi (with Replan)	0.0%	36.0%	0.0%	0.0%	6.7%	0.0%
AVDC	30.7%	28.0%	21.3%	8.0%	34.7%	17.3%
VLP	33.3%	28.0%	17.3%	8.0%	36.0%	18.7%
Diffusion Policy	45.3%	45.3%	8.0%	0.0%	40.0%	18.7%
AVDC-Replan	72.0%	89.3%	37.3%	18.7%	60.0%	24.0%
AVDC-IS-Replan	66.7%	93.3%	40.0%	21.3%	65.3%	29.3%
VideoAgent	40.0%	29.3%	13.3%	9.3%	38.7%	18.7%
VideoAgent (Iter2)	48.0%	40.0%	24.0%	12.0%	42.7%	36.0%
VideoAgent (Iter3)	58.7%	50.7%	28.0%	18.7%	53.3%	41.3%
VideoAgent-Replan	**82.7%**	**97.3%**	**40.0%**	**26.7%**	**73.3%**	**44.0%**

	faucet-close	faucet-open	handle-press	hammer	assembly	**Overall**
BC-Scratch	18.7%	22.7%	28.0%	0.0%	0.0%	15.4%
BC-R3M	18.7%	17.3%	37.3%	0.0%	1.3%	16.2%
UniPi (with Replan)	4.0%	9.3%	13.3%	4.0%	0.0%	6.1%
AVDC	12.0%	17.3%	41.3%	0.0%	5.3%	19.6%
VLP	30.7%	10.7%	33.3%	0.0%	1.3%	19.8%
Diffusion Policy	22.7%	**58.7%**	21.3%	4.0%	1.3%	24.1%
AVDC-Replan	53.3%	24.0%	81.3%	8.0%	6.7%	43.1%
AVDC-IS-Replan	48.0%	28.0%	78.7%	10.7%	0.0%	43.8%
VideoAgent	46.7%	12.0%	36.0%	0.0%	1.3%	22.3%
VideoAgent (Iter2)	53.3%	28.0%	52.0%	1.3%	5.3%	31.2%
VideoAgent (Iter3)	58.7%	36.0%	64.0%	1.3%	9.3%	38.2%
VideoAgent-Replan	**74.7%**	**46.7%**	**86.7%**	**8.0%**	**10.7%**	**53.7%**

Table 2.8: Meta-World Results. The mean success rates of baselines and VideoAgent on 11 simulated robot manipulation environments from Meta-World. VideoAgent consistently outperforms baselines across all tasks.

	door-open	door-close	basketball
AVDC	30.7%	28.0%	21.3%
VideoAgent	40.0%	29.3%	13.3%
VideoAgent-Online (Iter1)	48.0%	40.0%	24.0%
VideoAgent-Online (Iter2)	58.7%	50.7%	28.0%
VideoAgent-Online (Iter3)	58.7%	52.0%	26.7%

	shelf-place	btn-press	btn-press-top
AVDC	8.00%	34.7%	17.3%
VideoAgent	9.3%	38.7%	18.7%
VideoAgent-Online (Iter1)	12.0%	42.7%	36.0%
VideoAgent-Online (Iter2)	18.7%	53.3%	41.3%
VideoAgent-Online (Iter3)	17.3%	54.7%	40.0%

	faucet-close	faucet-open	handle-press
AVDC	12.0%	17.3%	41.3%
VideoAgent	46.7%	12.0%	36.0%
VideoAgent-Online (Iter1)	53.3%	28.0%	52.0%
VideoAgent-Online (Iter2)	58.7%	36.0%	64.0%
VideoAgent-Online (Iter3)	56.3%	36.0%	66.7%

	hammer	assembly	**Overall**
AVDC	0.00%	5.30%	19.6%
VideoAgent	0.0%	1.3%	22.3%
VideoAgent-Online (Iter1)	1.3%	5.3%	31.2%
VideoAgent-Online (Iter2)	1.3%	9.3%	38.2%
VideoAgent-Online (Iter3)	1.3%	10.7%	38.22%

Table 2.9: Meta-World Result. The mean success rates of VideoAgent combined with successive rounds of data collection via Online Iterations and Replan modules as compared to AVDC baseline.

IDENTIFICATION AND SUGGESTION

Task: You are a video reviewer tasked with evaluating a series of actions depicted through eight consecutive image uploads. These images together simulate a video. This task is structured as a few-shot learning exercise, where you will first review three examples and then apply learned principles to new queries.
Query-Evaluation Phase:

- **Inputs Provided:**
  - **Textual Prompt:** Describes the intended outcome or task the video aims to accomplish.
  - **Conditioning Image:** Establishes the fixed elements of the scene.
  - **Sequence of Images (7 Frames):** Illustrates consecutive moments in the video, representing the action sequence.
- **Evaluation Process:**
  - **Frame-by-Frame Analysis:** Carefully examine each of the seven images to understand the progression and continuity of actions.
  - **Assess Overall Coherence:** Evaluate the sequence as a whole to determine if the actions transition smoothly from one frame to the next while maintaining logical progression.
  - **Check for Physical Accuracy:** Ensure each frame complies with the laws of physics, identifying any discrepancies in movement or positioning.
  - **Verify Task Completion:** Confirm if the sequence as a whole accomplishes the task described in the textual prompt.
  - **Identify Inconsistencies:** Detect inconsistencies in object movement or overlaps that contradict the fixed scene elements depicted in the conditioning image.

IDENTIFICATION AND SUGGESTION (Continued)

Task: You are a video reviewer tasked with evaluating a series of actions depicted through eight consecutive image uploads. These images together simulate a video. This task is structured as a few-shot learning exercise, where you will first review three examples and then apply learned principles to new queries.
Query-Evaluation Phase:

- **Evaluation Criteria:**
  - **Descriptive Feedback:** Based on your evaluation, provide a concise, constructive sentence suggesting specific improvements. Focus on enhancing physical accuracy and task fulfillment based on identified inconsistencies or discrepancies.
- **Response Requirement:**
  - Feedback must be derived from your observations during the evaluation and not exceed 20 words.
- **Additional Notes:**
  - No further clarification can be requested.
  - Elements from the conditioning image must match those in each frame of the sequence.

2.11 Task Descriptions and In-Context Examples for VLM Feedback

TASK DESCRIPTION AND SUCCESS CRITERIA

- **door-open**: The robot arm has to open the door by using the door handle.
- **door-close**: The robot arm has to close the door by pushing the door or the handle.
- **basketball**: The robot arm has to pick up the basketball and take it above the hoop.
- **shelf-place**: The robot arm has to pick up the blue cube and place it on the shelf.
- **button-press**: The robot arm has to press the red button from the side by pushing it inside.
- **button-press-topdown**: The robot arm has to press the red button from the top by pushing it downward.
- **faucet-close**: The robot arm has to use the red faucet handle and turn it anti-clockwise.
- **faucet-open**: The robot arm has to use the red faucet handle and turn it clockwise.
- **handle-press**: The robot arm has to press the red handle downward.
- **hammer**: The robot arm has to grip and pick up the hammer with a red handle and hit the peg on the box inside.
- **assembly**: The robot arm has to pick up the ring and place it into the red peg.

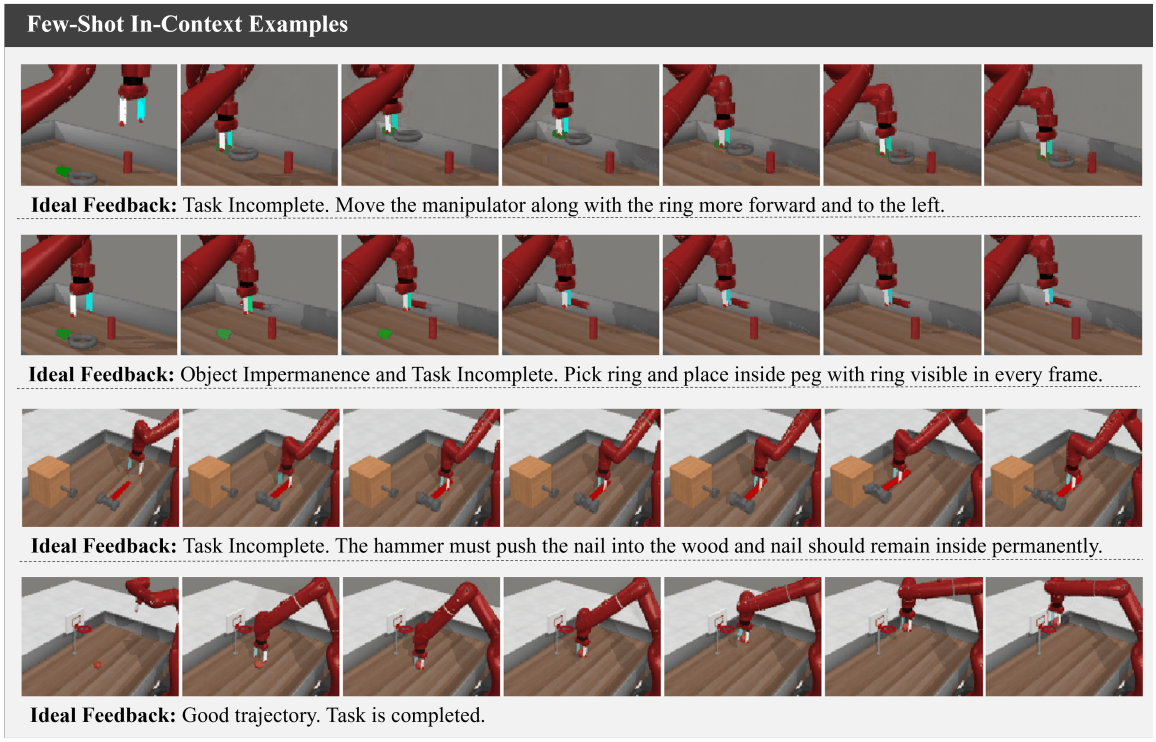


Figure 2.6: Few-Shot Examples given to VLM: We provide some examples to the VLM and corresponding feedback to teach the VLM in-context how to critic the generated videos for task completion and success or failure.

2.12 Examples

Additional examples are provided below.

2.12.1 Zero-shot generalization on real-world scenes

VideoAgent trained on Bridge dataset demonstrates strong performance on zero shot video generation for natural distribution shifts and longer language instructions. Some examples of the synthesized videos can be found in Fig. 2.7.

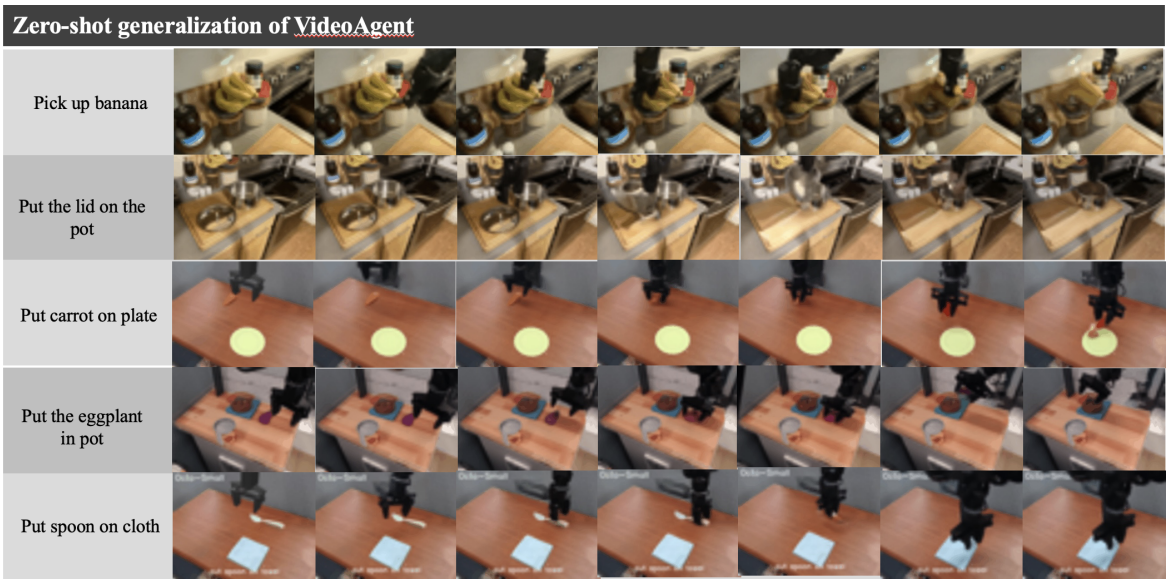


Figure 2.7: Zero-shot generalization of VideoAgent: VideoAgent generalizes fairly well to natural distribution shifts and is able to generate successful trajectories on data it has not been trained on.

2.12.2 Improvements on real-world scenes

We show an example video with and without refinement in Figure 2.8, where the baseline (middle row) hallucinates (the bowl disappears) whereas VideoAgent produces the video that completes the task (bottom row).

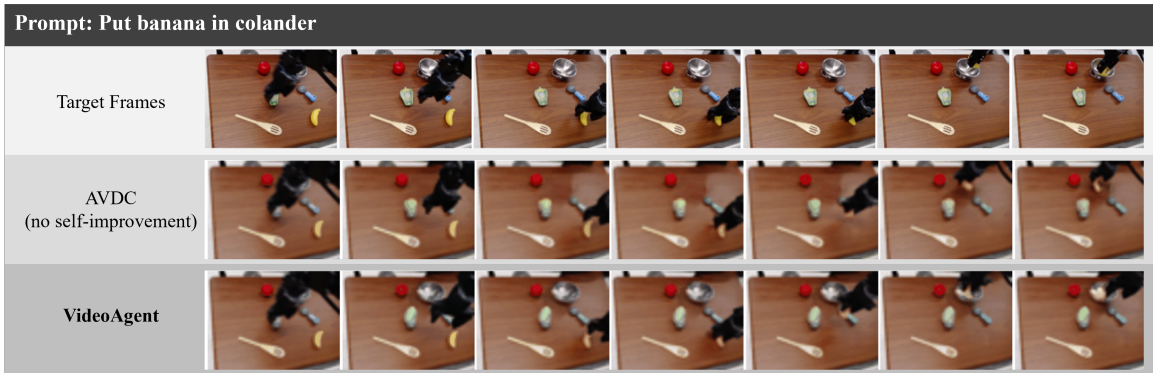


Figure 2.8: Correcting Hallucinations in Video Generation: The AVDC model hallucinates after the second frame, and totally removes the colander from the frames and places the banana on the table. In contrast, VideoAgent accurately retains the colander's position and correctly places the banana inside.

2.12.3 Improvements in Meta-World

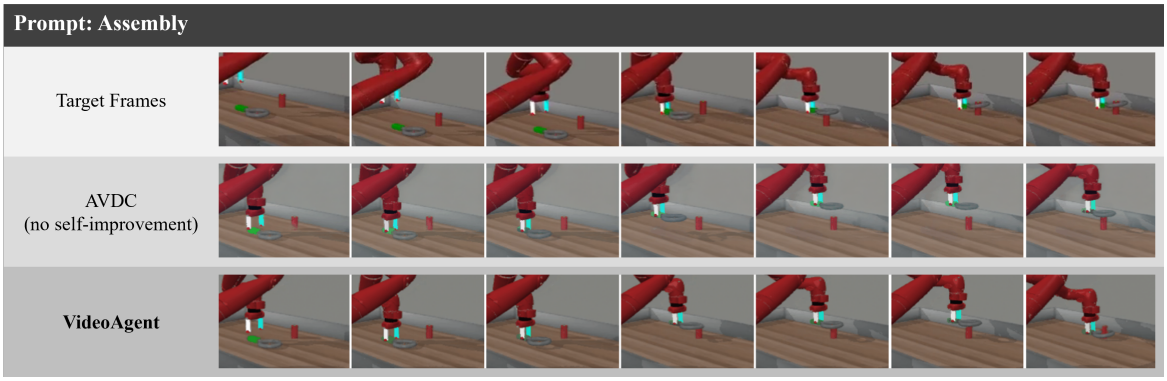


Figure 2.9: Correcting Hallucinations in Video Generation: The goal prompt is "Assembly" as shown in the Target Video. The AVDC model has problem of object permanence and action incomplete in last frame. In contrast, our VideoAgent model accurately object permanence and correctly places the inside the peg properly.

2.12.4 Improvements in iThor

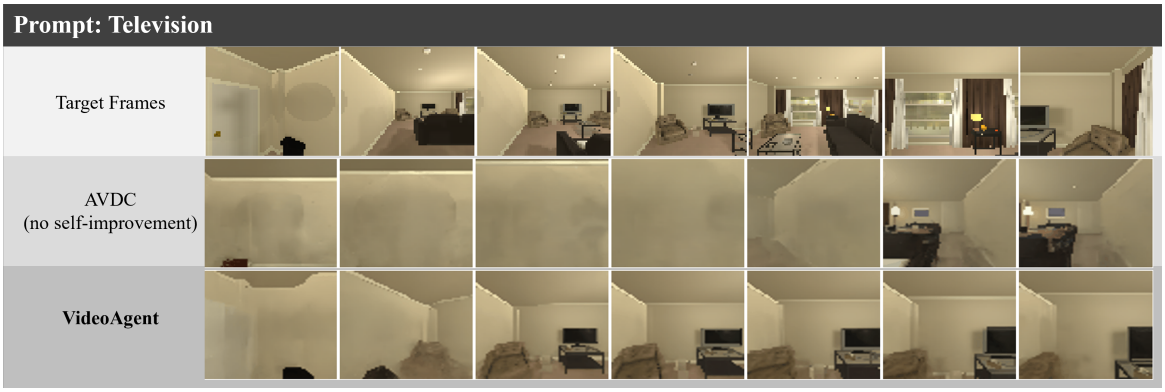


Figure 2.10: Correcting Hallucinations in Video Generation: The goal prompt is "Television" as shown in the Target Video, the goal is for the navigator to locate the object and reach near it. The AVDC model has difficulty reconstructing and navigating in the livingroom to find the television. In contrast, our VideoAgent model solves the initial frame hallucinations and accurately reaches near the television correctly.

2.12.5 Identification and Suggestive Feedback Examples

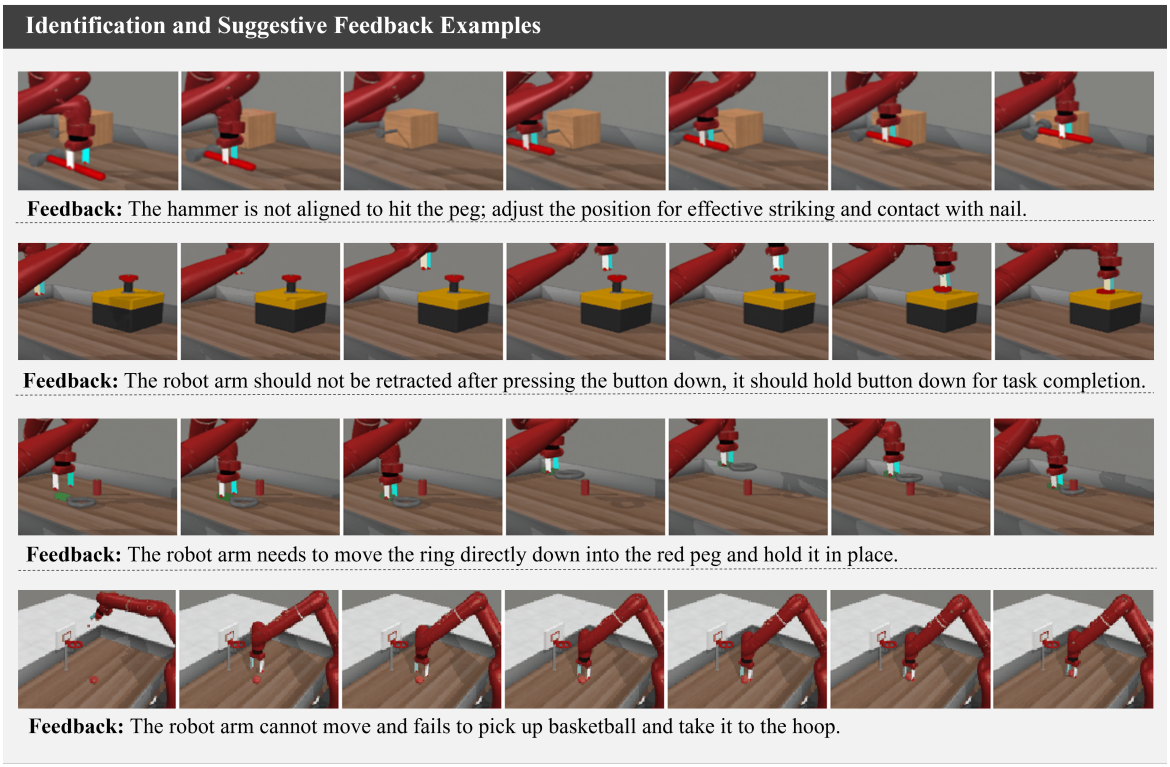


Figure 2.11: Detailed VLM Feedback: We show the efficacy of VLMs to provide useful feedback even in the absence of access to a simulator or real-world execution environment. The VLM acts as a proxy reward model to condition VideoAgent on useful corrective signals, leading to improved performance as described in Table 2.3.

2.13 Conclusion, Limitations and Future Work

We have presented VideoAgent , where a video generation model acts as an agent by generating and refining video plans, converting video plans into actions, executing the actions in an environment, and collecting additional data for further self improvement. Through interaction with an external environment, VideoAgent provides a promising direction for grounding video generation in the real world, thereby reducing hallucination and unrealistic physics in the generated videos according to real-world feedback.

Limitations and Future Work. VideoAgent needs to overcome a few limitations that I discuss below.

- In the online setting, VideoAgent only considers filtering for successful trajectories for further finetuning. Exploring other algorithms such as online RL is interesting future work.
- VideoAgent utilizes optical flow for action extraction. It would be interesting to see how VideoAgent works with inverse dynamics model or image-goal conditioned diffusion policy.
- We only measured end-to-end task success in simulated robotic evaluation settings. It would be interesting to see how VideoAgent works with real robotic systems.
- As additional data is being collected in the online setting, in addition to finetuning the video prediction model, one can also finetune the action extraction module (flow model), and the VLM feedback model using the additionally collected data, which we defer to future work.
- VideoAgent trades off inference-time compute for better performance by iteratively refines generated video plans under the guidance of a VLM. Exploring other inference-time search strategies to further improve video sample quality is an interesting area of research.

Chapter 3

Beyond Rule-Based and Human Verifiers: Scaling Reasoning Training Compute with Unverified Synthetic Data and Reward Free Methods

3.1 Introduction

Recent advancements in enhancing large language models (LLMs) for reasoning tasks often utilize meticulously verified and filtered reasoning data and rule-based verifiers, which are constrained by the availability of ground-truth solutions for a fixed set of problems. In contrast, moving towards an unsupervised skill learning setup eliminates the dependence on a fixed verifiable problem set, expanding the horizon to use diverse synthetic problems as well as the use of LLM-based verifiers, offer the potential to apply verification to reasoning tasks lacking ground-truth solutions. While recent works on improving reasoning guided by RL finetuning have become popular, it is again bottlenecked by rule-based verifiers and labeled chain-of-thoughts (CoTs). Moreover, recent investigations have shown that an efficient way to elicit such capabilities, especially for smaller models, is to distill from a teacher model, i.e. supervised-finetuning (SFT) on reasoning traces output by a stronger reasoning model. Even for larger models, SFT with thousands of examples typically serves as a crucial precursor to RL [73, 19]. This reliance on distillation is underscored by findings that RL on its own does not effectively enhance a model's reasoning priors [249], whereas SFT enables the introduction of novel reasoning primitives that RL can then refine. Moreover,

recent works also show that RL fine-tuning, even with spurious rewards [180], leads to performance gains, further necessitating the study of a data-centric approach towards the reasoning CoTs and capabilities in the base model that lead to acquiring downstream reasoning capabilities. In our work, we focus on studying this in detail, along with scaling trends of synthetic data - both problems and solutions - generated by LLMs and especially the role of how wrong CoTs can also help in such circumstances counterintuitively. We investigate the impact of incorporating unfiltered synthetic data into model training to push performance and present a framework to understand these performance gains. Additionally, we perform detailed ablations on the MATH [79] benchmark to demonstrate our findings on three classes of model families - Gemma 2, Llama 3.1, and Qwen 2.5.

Open-source community efforts such as Open-R1 [95] and OpenThoughts [72] have focused on harvesting high-quality filtered CoT trajectories from strong reasoning models. More recent approaches, such as LIMO [242] and S1K [147], emphasize the value of selecting high-quality CoT traces for distillation. They demonstrate that as few as 1,000 carefully chosen examples can significantly enhance a student model's performance, particularly on mathematical reasoning tasks [242, 147]. However, both LIMO and S1K restrict their curated datasets to easily verifiable domains.

Our work presents a counter-intuitive yet powerful hypothesis: LLMs do not require perfectly clean data to learn how to reason. We propose that models can acquire and refine robust reasoning abilities by training on a large corpus of imperfect CoT demonstrations, including those that contain errors and lead to incorrect final answers. This hypothesis is predicated on the observation that even flawed reasoning attempts are not random noise; they often contain valid sub-sequences of logic, correct application of formulas, or sound decompositions of a problem. These nuggets of correct reasoning represent a valuable, albeit noisy, learning signal.

The inspiration for applying this lens of analysis in LLMs is drawn from a compelling analogy in cognitive science: the process of human language acquisition. Decades of research have shown that children acquire robust, systematic language grammars even when exposed to inconsistent, error-filled, and probabilistic linguistic input from their environment, such as from non-native speaking parents. A seminal case study by [186] documented a deaf child, Simon, who learned American Sign Language from parents who were late, non-native signers. Simon's linguistic output was more grammatically regular and consistent than the noisy input he received; he did not simply mimic his parents' errors but instead appeared to regularize the input, filtering out inconsistencies and amplifying the underlying grammatical patterns [186]. Subsequent experimental work by [12] confirmed this phenomenon, showing that young children regularize inconsistent linguistic input by latching onto and amplifying the most dominant, consistent forms, a stark contrast to adults who tend to merely match

the probabilities of the input they observe [12, 58].

We argue that taking inspiration from learning in children from imperfect teachers, we develop **Regularized Reasoning Bootstrapping (Reg-ReBoot)**, which can be harnessed to train reasoning LLMs. The core idead is to utilize the model's training process that naturally emphasizes the consistent, underlying patterns in the reasoning data, effectively down-weighting or averaging out the diverse, idiosyncratic noise introduced by individual errors.

Our contributions in this work are threefold:

1. A Data-centric Framework to Learning from Unverified CoTs: We introduce and formalize Reg-ReBoot, a finetuning paradigm that leverages large-scale, unverified, and imperfect CoT data to enhance LLM reasoning, thereby circumventing the primary bottleneck of existing method's dependence on verifiers and high-quality filtered reasoning data.

2. Analysis and Grounding of Reg-ReBoot from Learning over Noisy Data: We provide a detailed analysis that connects the standard maximum likelihood training objective of LLMs to the cognitive concepts of regularization and frequency boosting common in human and child learning. We demonstrate how this objective, when applied to noisy data, naturally gives rise to a preference for consistent reasoning primitives, an effect underpinned by the known inductive biases of large neural networks.

3. Empirical Validation and Ablations: We present detailed experimental results in the mathematical reasoning tasks, using both knowledge distillation from a stronger teacher model and self-distillation. Our results on Gemma-2-2B and Llama-3.1-8B models show that fine-tuning on datasets composed entirely of CoTs with incorrect final answers can lead to substantial performance gains in non-reasoning base models, rivaling and surpassing the effectiveness of training on human-curated gold datasets.

Our work tries to present a framework towards a more scalable and practical approach for developing advanced reasoning in LLMs, one that embraces the noisy, imperfect nature of real-world data rather than depending on an idealized, and largely unattainable, standard of perfect supervision. This investigation underscores a key departure from prior work that insisted on perfectly verified solutions – our method can exploit noisy, unverified data, but only insofar as that data contains some consistent reasoning patterns. Understanding the limits of this assumption will inform how broadly Reg-ReBoot can be applied, especially in domains where obtaining fully correct solutions is impractical.

3.2 Related Work

Our work builds on research situated at the confluence of learnability in large models, learning from noisy data and the notions of leveraging such ideas on unsupervised unverified reasoning data for improving LLMs.

Data-Centric Approaches for Enhancing Reasoning

Distillation and reinforcement learning (RL) have emerged as two foundational strategies for building models with strong reasoning abilities. Notably, [73] present a systematic comparison between these approaches, demonstrating that distillation from a powerful teacher model substantially outperforms RL training conducted solely on the student model. Following this, the research community has devoted significant attention to distillation-based techniques, leading to the release of big high-quality reasoning datasets such as OpenThoughts [72] and OpenR1 [95]. Studies like LIMO [242] and S1K [147] highlight that well-designed questions and reasoning traces can drastically improve sample efficiency - showing that as few as 1000 examples can be sufficient to transfer complex chain-of-thought (CoT) reasoning patterns from teacher to student models. However, these findings have largely been confined to domains and datasets with easily available verifiers. In contrast, our work offers new perspectives by learning from data-centric distillation strategies across a range of unverified data posing the challenge of learning reasoning for imperfect CoTs.

Chain-of-Thought Reasoning and Knowledge Distillation from Large Models

Chain-of-thought (CoT) prompting, encouraging language models to articulate intermediate reasoning steps - has emerged as a highly effective technique for solving complex problems [225]. Both few-shot CoT [63] and zero-shot CoT [111] evoke the latent reasoning abilities of large language models by guiding them through incremental sub-problems. Building upon this, Least-to-Most prompting [264] explicitly breaks down tasks into simpler subtasks, solving them sequentially to tackle complexity. Self-Consistency [218] further enhances performance by aggregating multiple diverse reasoning paths and selecting the most consistent answer. Progressive-Hint Prompting (PHP) [260] refines this process by iteratively using previously generated answers as additional context, enabling convergence towards the correct solution. This iterative framework and majority voting based test-time methods create soft filters to filter correct from wrong.

We instead focus on studying how even incorrect responses from LLMs can serve as valuable knowledge, we argue that rationales themselves, not just final answers, contain

instructive signals, and we incorporate these negative rationales directly into the knowledge transfer process during student model training, rather than solely using them during inference. Our framework currently works best under the assumption that the synthetic CoT data contain some underlying correct reasoning structure rather than being entirely random. In practice, we expect imperfect solutions to still include valid intermediate steps ('static noise' as opposed to completely derailing errors). As an ongoing extension of this work, we are conducting controlled experiments to validate this assumption. In these experiments, we gradually inject increasing levels of noise or wrong steps into otherwise correct CoTs – starting from fully correct human-written solutions and progressively introducing errors – to observe at what point the model's performance significantly degrades. This will help quantify how tolerant our approach is to noise and confirm that Reg-ReBoot works because of partial correctness in the training data. Our goal is to rigorously determine the boundary where the learning signal from coherent sub-steps is overcome by too much random noise. By doing so, we address the concern of CoT structure: we show that as long as a certain fraction of each reasoning trace is logically sound, the model can bootstrap its reasoning abilities from it.

Knowledge distillation (KD) is a proven method for transferring knowledge from a high-capacity teacher model to a smaller student, typically by minimizing divergence between their output distributions, hidden states, or attention maps [85, 175, 59, 100, 203]. However, with mainstream LLMs (e.g., GPT-4) operating as black boxes, traditional KD methods can be hard to implement. As a result, researchers have turned to off-policy distillation by fine-tuning student models directly on LLM-generated responses with correct answers [88, 63, 268]. Yet, this discards LLM outputs that contain incorrect final answers, despite the potential value embedded in their latent reasoning. Considering that a sizable fraction of LLM-generated reasoning is inaccurate in complex tasks, our work introduces a framework to leverage this abandoned knowledge, integrating both correct and incorrect CoTs into the distillation process to transfer reasoning capabilities more effectively.

Learning from Noisy Signals - From Human Learners to LLMs

Deep neural networks—including large language models (LLMs)—exhibit a well-documented inductive bias: they first internalize simple, low-frequency regularities before over-fitting idiosyncratic noise [10, 11, 167]. Recent work extends this insight to pretraining data in LLMs. [76] distinguish static noise - localized slips such as a single miscalculation—from dynamic noise, logical errors that corrupt all subsequent steps, and show that LLMs remain remarkably robust to large amounts of the former yet are highly sensitive to even small amounts of the latter. Our empirical findings mirror this pattern: the wrong CoTs that yield performance gains in our experiments mostly contain static noise, enabling the model

to absorb the overarching reasoning template while ignoring minor slips.

This phenomenon also appears in human language acquisition. Children confronted with variable or inconsistent input actively regularize, amplifying the most consistent pattern and suppressing irregular variants. The case of Simon—a deaf child who produced more grammatical American Sign Language (ASL) than the inconsistent input provided by his non-native-signer parents—exemplifies this frequency-boosting mechanism [186]. Laboratory studies with artificial languages corroborate that learners regularize unpredictable variation while faithfully reproducing lexically consistent alternations, and that relative dominance, not absolute frequency, drives this process [12, 58]. Together, evidence from neural networks and human learners points to a shared inductive bias: when faced with noisy or inconsistent data, both systems preferentially internalize dominant, systematic structure and filter out erratic noise - a principle that underlies the success of our data-centric approach.

While our LLM is obviously not 'learning' in the human cognitive sense, its gradient-based training objective produces a similar effect – amplifying consistent reasoning patterns and suppressing irregular deviations – much like how children regularize noisy language input.

3.3 Method and Analysis: Regularized Reasoning Bootstrapping (Reg-ReBoot)

Most recent works in improving LLM reasoning depend on two primary techniques - either using a carefully curated filtered dataset of CoTs using heuristic methods or using a reward model / verifier to filter for SFT runs and rank generations for RL training. Our primary finding in this work is that non-reasoning LLMs with broadly general base capabilities, can learn robust reasoning skills from imperfect Chain-of-Thought (CoT) without any explicit filtering or verification steps, because these examples, even when incorrect, contain partially correct reasoning structures. In this section, we formalize this hypothesis, drawing inspiration from the principles of human cognitive development and connecting them to the mechanics of statistical learning in neural networks and LLMs. We propose a framework, Reg-ReBoot, to explain how a model trained from an imperfect teacher on a mixture of correct and incorrect CoTs can converge toward useful, generalizable reasoning patterns.

3.3.1 The Latent Structure of Reasoning: Primitives and Traces

To formalize our analysis, we first introduce a conceptual model for the structure of reasoning. We posit that any given CoT is a surface-level linguistic realization of a deeper, latent structure. Let a problem be denoted by $x \in \mathcal{X}$. A CoT solution, y, is a sequence of tokens

$y = (t_1, t_2, \ldots, t_L)$ that represents a step-by-step solution to x. We model this sequence y as being generated from an underlying latent reasoning trace, $Z = (s_1, s_2, \ldots, s_T)$, where each s_i is a reasoning primitive.

A reasoning primitive s_i represents a fundamental, atomic unit of thought or computation. For a mathematical problem, a primitive could be an arithmetic operation, the application of a specific formula, the definition of a variable, or a step of logical deduction. Crucially, each primitive s_i in a trace possesses a latent correctness attribute, $c_i \in \{0, 1\}$, which depends on the context established by the preceding primitives $Z_{<i} = (s_1, \ldots, s_{i-1})$. A value of $c_i = 1$ signifies that the primitive is a valid and correct step given the context, while $c_i = 0$ indicates an error. A perfectly correct CoT, denoted y^*, corresponds to a latent trace Z^* where all primitives are correct ($c_i = 1$ for all i). An incorrect CoT, y_e, corresponds to a trace Z_e that contains at least one incorrect primitive ($c_i = 0$ for some i).

3.3.2 Modeling Imperfect Demonstrations from a Noisy Teacher

In our experimental setup, we do not have access to a dataset composed solely of perfect demonstrations y^* and intentionally avoid using compute intensive filtering and verification steps to study the scaling trends of this phenomenon of learning from imperfect reasoning CoTs. Instead, we train on these noisy data generated by a model, which could be a more powerful LLM (knowledge distillation from teacher) or the model itself (self-distillation). This data is a mixture of correct and incorrect CoTs. We model the training data distribution for a given problem x as a mixture:

$$P_{\text{mix}}(y|x) = \lambda P_c(y|x) + (1 - \lambda) P_e(y|x)$$

where $\lambda \in [0, 1]$ is the proportion of correct examples.

- $P_c(y|x)$ is the distribution over correct CoTs, y^*. For a given problem, we can assume this distribution is sharply peaked around the ideal solution trace.
- $P_e(y|x)$ is the distribution over incorrect CoTs, y_e. This is the critical component. An incorrect CoT is not random noise. Its latent trace Z_e is a sequence of primitives where some are incorrect ($c_i = 0$), but many others may still be correct ($c_j = 1$).

Drawing on the analysis of [76], where they studied the effect of noise in pretraining, we can further classify the errors within P_e:

- **Static Errors:** These are local mistakes that affect a single primitive s_i but do not corrupt the high-level structure of the subsequent reasoning trace. A simple

arithmetic mistake is a canonical example. The rest of the reasoning can proceed logically, albeit from an incorrect intermediate value.

- **Dynamic Errors:** These are more fundamental mistakes that corrupt the state of the reasoning process, rendering all subsequent primitives invalid in that context. Choosing the wrong formula or making a critical logical fallacy at an early step are examples of dynamic errors.

Our manual analysis of solution CoTs of 100 random questions from the Hendricks Math training dataset questions generated using the gemma27B model, which serves as a primary source for our noisy teacher, indicates that a significant fraction (68%) of the CoTs with incorrect final answer, suffer from what can be categorized as static errors - minor calculation mistakes or skipped steps leading to incorrect final answer, but leaving the overarching reasoning approach intact. This suggests our training data is rich in traces that are locally flawed but globally structured, a condition under which models have been shown to be robust learners of reasoning skills and primitives. This idea of student model regularizing over dynamic errors to learn useful concepts has also been extensively studied in children language acquisition and sign language acquisition studies, providing the inspiration to pursue this framework for analyzing such behaviours in our current class of reasoning LLMs as well[186, 12].

3.3.3 The Implicit Regularization of Maximum Likelihood Learning

The student model, with parameters θ, is trained to maximize the likelihood of the observed data. This corresponds to minimizing the standard negative log-likelihood (NLL) loss, or cross-entropy loss, over the mixture distribution:

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y)\sim P_{\text{mix}}} \left[-\log P_\theta(y \mid x) \right]$$

Expanding this for a sequence $y = (t_1, \ldots, t_L)$, the loss becomes a sum over the conditional probabilities of each token:

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y)\sim P_{\text{mix}}} \left[\sum_{k=1}^{L} -\log P_\theta(t_k \mid x, t_{<k}) \right]$$

We analyze how the optimization process behaves under this objective when the data is a mixture of correct and flawed reasoning traces.

1. Consistent Signal from Correct Primitives Consider a prefix of tokens $t_{<k}$ that corresponds to a correct partial reasoning trace $Z^*_{<j}$. The next step in a correct solution involves generating tokens corresponding to the correct primitive s^*_j. This correct continuation is present in 100% of the samples drawn from the correct distribution P_c. Furthermore, it is also present in a large fraction of the samples from the error distribution P_e—specifically, all those traces that have not yet deviated due to a dynamic error. Therefore, the gradient signal $\nabla_\theta \log P_\theta(t_k|x, t_{<k})$ for the tokens representing s^*_j is strong and, crucially, consistent across a large portion of the training batch. The optimization process receives a clear, unambiguous signal to increase the probability of this correct reasoning step.

2. Inconsistent Signal from Error Primitives Now, consider a prefix $t_{<k'}$ that has led to an error state (i.e., the last primitive was incorrect). The subsequent reasoning paths become highly divergent. One erroneous trace might continue with a specific type of calculation error, another might make a different calculation error, a third might apply an incorrect logical rule, and a fourth might attempt to backtrack. From this single erroneous context $t_{<k'}$, the model observes a wide variety of possible next tokens $t_{k'}$. The gradient signal for any single specific error path is diluted and weak, as it is averaged with the competing signals from all other possible error paths.

The net effect of this dynamic is that the learning process, driven by gradient descent, naturally amplifies the probability of the consistent, correct reasoning primitives that are shared across both fully correct and partially correct traces. Simultaneously, it effectively treats the diverse, idiosyncratic error patterns as low-signal noise to be averaged out and suppressed.

3.3.4 Formalizing the Analogy to Cognitive Regularization

This learning dynamic is not a coincidence; it is a direct consequence of the inherent properties of both the learning algorithm and the data structure, and it provides a formal bridge to the concept of regularization observed in human cognitive development. Children exposed to probabilistic grammars tend to over-regularize, choosing the most frequent form and ignoring less frequent alternatives [94, 186]. LLMs appear to exhibit similar behavior during training - favoring majority-consistent transitions even when trained on noisy data.

Inductive Bias towards Simplicity The behavior described above is a manifestation of the well-established **inductive bias** of deep neural networks. When over-parameterized, these models do not learn by rote memorization. We also performed experiments to see the effect of overfitting on wrong CoTs and current LLMs do not operate in the overfitting and memorization regime, thereby enabling these models to learn useful reasoning skills

even from partially wrong CoTs due to the ability of the models existing inductive biases enabling it to help regularize over the errors in the data. Instead, they exhibit a preference for learning simpler functions that explain the data [11, 9]. In our context, the set of correct reasoning primitives and their compositional rules forms a simple low-entropy, and coherent system. In contrast, the universe of possible errors is vast, unstructured, and high-entropy. The model's inductive bias pushes it to first learn the underlying basics of reasoning before it begins to memorize the noisy exceptions (the errors).

Frequency Boosting as Conditional Probability Maximization The frequency boosting mechanism observed in previous mentioned works in child language acquisition [186], can be formalized within this framework. At any reasoning step j, given a correct context (prefix) $Z^*_{<j}$, the training data provides an empirical distribution over the set of possible next primitives, $P_{\text{data}}(s_j|Z^*_{<j})$. In this distribution, the correct primitive s^*_j occurs with some probability p^*, while a set of various incorrect primitives $\{s_{j,e_1}, s_{j,e_2}, \dots\}$ occur with lower probabilities $\{p_{e_1}, p_{e_2}, \dots\}$.

Inspired by the findings of [12, 58], we see that effective learning can proceed as long as the correct primitive is **relatively dominant**, meaning its probability is greater than that of any single incorrect alternative: $p^* > \max_k(p_{e_k})$. The NLL objective function encourages the model's predictive distribution $P_\theta(s_j|Z^*_{<j})$ to match the empirical distribution P_{data}. However, due to the model's finite capacity and its inductive bias towards learned reasoning primitives during pretraining, it will not perfectly replicate this noisy distribution. Instead, it will allocate a disproportionately large probability mass to the mode of the distribution, which is the correct and most consistent primitive, s^*_j.

This is the mathematical analogue of cognitive regularization in LLM. The model does not probability-match the noisy input. Instead, it boosts the probability of the most consistent and systematically occurring option. By iteratively applying this principle at each step, the model learns to stitch together a complete and correct reasoning trace, effectively bootstrapping its way to competence from a foundation of imperfect demonstrations without the need of explicit verifiers or data filtering in the mid-training stage.

3.3.5 Distillation for Reasoning

The fundamental learning method we adopt in this work is knowledge distillation, a process where a smaller student model which in our case is our base non-reasoning LLMs are trained to mimic the reasoning trajectories of a larger more capable teacher model [84]. Recent works have already shown the efficacy of distillation as an effective if not better and more efficient method to improve reasoning in smaller LLMs compared to RL finetuning [249, 19]. This allows us to transfer the powerful reasoning capabilities of a frontier model to a smaller,

more efficient one. Our work investigates distillation as a purely off-policy learning problem, where the student learns from a static dataset of reasoning chains generated by a fixed policy teacher model.

The motivation for this strictly off-policy approach is to isolate the learning process from any form of online verification or feedback. In many contemporary methods, a student model's generations are evaluated or guided by a reward model or a more powerful teacher model, which then provides a signal to update the student's policy (an on-policy loop). We deliberately avoid such on-policy or online distillation schemes to better study our verification free learning setup. Our goal is to understand if a student model can improve by simply observing and imitating complete and noisy reasoning trajectories without any explicit labels and without being guided token-by-token or sequence-by-sequence by an external verifier. We achieve this by treating all generated reasoning chains—whether from a powerful teacher or from the student model itself - as a static data buffer. The student then learns from this fixed dataset via standard supervised fine-tuning (SFT).

In this off-policy framework, we explore two sources for the training data buffer:

First, in a conventional cross-model setup, we leverage a powerful teacher model, P_T, to generate a large, synthetic dataset of reasoning trajectories, $\mathcal{D}_T$. The student model, $P_S(\theta)$, learns from this fixed dataset. The goal is to train the student's parameters θ by minimizing the cross-entropy loss for each token in a generated chain-of-thought (CoT) sequence $y = (y_1, \ldots, y_T)$ given a problem x. The distillation loss, $\mathcal{L}_{KD}$, for a single trajectory (x, y) from the teacher's dataset $\mathcal{D}_T$ is given by:

$$\mathcal{L}_{KD}(\theta) = -\sum_{t=1}^{T} \log P_S(y_t | x, y_{<t}; \theta)$$

Crucially, as our work demonstrates, the teacher-generated CoT, y, does not need to be correct for learning to occur. This is a significant finding, as it suggests the structural patterns of reasoning are themselves a valuable learning signal, independent of the final answer's correctness. This is a significant departure from traditional distillation works, which typically assumes high-quality or perfect data from teacher.

Second, we also test this off-policy setup on self-distillation. Here, the base student model refines its reasoning abilities bootstrapped by its own CoTs. We use the current student model $P_S(\theta_k)$ as a fixed policy to generate a new dataset of CoT solutions, $\mathcal{D}_{S,k}$. This dataset is then treated as a static data buffer for the next stage of reasoning fine-tuning. This process is still off-policy because the model being trained (θ_{k+1}) learns from a fixed

distribution created by a previous, separate policy (θ_k). The update rule is expressed as:

$$\theta_{k+1} = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}_{S,k}} \left[-\sum_{t=1}^{T} \log P_S(y_t|x, y_{<t}; \theta) \right]$$

where the dataset $\mathcal{D}_{S,k}$ is generated by sampling from $P_S(\cdot|x; \theta_k)$. We stick to a single update loop (k =0), although previous works have shown in detail how multiple rounds of finetuning on data generated by the iteratively improving base model is a successful recipe to further push performance as well. This self-distillation loop, free from any external verification, allows the model to improve by learning from its own generated reasoning structures. Our experiments show that fine-tuning on these self-generated, and often incorrect, CoTs still leads to significant performance improvements.

3.4 Experiments and Results

To study the effect of learning dynamics under our experimental setup of inconsistent noisy reasoning inputs, we perform a detailed study using Supervised Fine-Tuning (SFT) as the mid-training paradigm to enable non-reasoning models to acquire reasoning skills by Reg-ReBoot. Recent works have shown that for small base models SFT is just as effective and often outperforms RL finetuning at much less compute resources and data. We study the effect of noisy CoTs on three base models namely Gemma 2-2 B, Llama 3.1-8 B, and Qwen 2.5-1.5 B. And as for the data we rely on the Hendrycks MATH data's training set and generated synthetic solutions for our controlled SFT experiments and on MATH128 which is a harder subset of the Hendrycks MATH test set for our evaluations [80].

3.4.1 Experimental Setup

Models The experiments utilized three prominent open-source non-reasoning LLMs as base models for fine-tuning: Gemma-2-2B, Llama-3.1-8B, and Qwen 2.5-1.5B. We chose these models to study the effect of reasoning acquisition across model families and the generality of such methods.

Datasets with MATH training set problem prompts All fine-tuning datasets were derived from mathematical problems, primarily from the MATH dataset [80]. Here we use both the human-generated gold solutions present in the original MATH 7500 training set as well as teacher model and self-generated synthetic solutions for studying the teacher distillation and self distillation setups. The key datasets used are:

- human_gold: The high-quality original dataset of 7,500 human-written, verified correct CoT solutions for problems in the MATH training set.
- gemma27B_gold: A dataset of CoT solutions generated by a powerful teacher model, Gemma 2-27B, and filtered to include only those that result in the correct final answer. The problem prompts are same as the MATH training set problems.
- gemma27B_wrong: A dataset of CoT solutions generated by the same Gemma 2-27B teacher model, but filtered to include only those that result in an incorrect final answer. This is our primary dataset for testing the Reg-ReBoot hypothesis. The problem prompts are same as the MATH training set problems.
- qwen14B_gold: A dataset of CoT solutions generated by a powerful teacher model, Qwen 2.5-14B, and filtered to include only those that result in the correct final answer. The problem prompts are same as the MATH training set problems.
- qwen14B_wrong: A dataset of CoT solutions generated by a powerful teacher model, Qwen 2.5-14B, but filtered to include only those that result in an incorrect final answer. The problem prompts are same as the MATH training set problems.
- self_distill_gold: Correct CoT solutions generated by the base models themselves, all on the same set of MATH training set problems.
- self_distill_wrong: Incorrect CoT solutions generated by the base models themselves, all on the same set of MATH training set problems.

We chose mathematics-related datasets as the primary test bed for our method for several reasons. Since Reg-ReBoot aims to improve latent reasoning capabilities in a skill-learning setup, domains like mathematics are a natural fit - the chain-of-thought (CoT) solutions in such domains often have well-defined, compositional subcomponents. Even if the complete CoT leads to an incorrect final answer, many intermediate steps may still be correct, providing useful learning signals. This property allows the model to learn and generalize from partially correct reasoning traces without requiring fully verified solutions. In contrast, domains where reasoning traces are less structured or harder to decompose (e.g., open-ended creative writing) offer less obvious latent reasoning substrates, making direct application of our current approach challenging. Also since base models are already adept with such general language generation tasks, gains in performance using methods like Reg-ReBoot will also be moderate.

We use the following standard zero-shot prompt template both during SFT and for all evaluations :

MATHEMATICAL PROBLEM SOLVING (ZERO-SHOT)

Zero-Shot Prompt Template :

> You are a math expert. I am going to give you a math Problem that you need to solve. When you answer, respond only with the Solution, thinking step by step. At the end of the Solution, when you give your final answer, write it in the form "Final Answer: The final answer is {answer}. I hope it is correct."
>
> Problem: {problem}
>
> Solution:

3.4.2 Details of SFT

For SFT, we finetune the model on teacher-generated CoTs while masking the loss on the question portion of the input, ensuring that only the model's reasoning trace and final answer contribute to the gradient. We set a maximum input length of 1024 tokens and ensure all completions fit within this budget. Training is conducted using the standard TRL SFTTrainer library built on HuggingFace Transformers. We train for 10 epochs on all datasets, using a per-device batch size of 2 with gradient accumulation of 128 steps when training on a single A100 40 GB GPU and a per-device batch size of 1 with gradient accumulation of 64 steps when training on a node of 4 A100 40 GB GPUs, effectively simulating a batch size of 256. We use the AdamW optimizer with 0.05 weight decay, a constant learning rate of $1e^{-6}$, and a warmup ratio of 0.1. The training is performed using BF16 precision and checkpointed every 10 steps for most runs.

3.4.3 Primary Findings: Learning to Reason via Distillation and from Unverified CoTs

Our central experimental finding is that off-policy distillation via supervised fine-tuning can turn small non-reasoning base LLMs into competent reasoners on small reasoning datasets after very efficient training runs. And the counterintuitive result we noticed is that datasets composed entirely of incorrect CoTs can lead to significant improvements in reasoning performance as well, challenging the conventional wisdom that only verified, correct demonstrations are useful for training. This opens up various opportunities to leverage lots of unverified synthetic data and removes the bottleneck of painstakingly curating verified CoT datasets.

Table 3.1: Peak SFT performance of Gemma 2-2B and Llama 3.1-8B on the MATH-128 test set.

Training Dataset	Peak SFT Accuracy
Gemma-2-2B	
Gemma-2-2B baseline	14.06%
human_gold	20.31%
gemma27B_gold	28.12%
gemma27B_wrong	27.34%
qwen14B_gold	22.65%
qwen14B_wrong	21.87%
self_distill_gold	21.87%
self_distill_wrong	21.87%
Llama-3.1-8B	
Llama-3.1-8B baseline	14.84%
human_gold	16.40%
gemma27B_gold	21.87%
gemma27B_wrong	27.34%
self_distill_gold	25.78%
self_distill_wrong	24.22%

Table 3.1 summarizes the peak accuracies achieved on the MATH-128 test set after SFT on the various datasets. The results demonstrate a consistent trend across both the Gemma-2-2B and Llama-3.1-8B models.

Analyzing the reasoning performance of Gemma 2-2B and Llama 3.1-8B The most significant result from these experiments is that fine-tuning on gemma27B_wrong achieves accuracy that is at par with that achieved by gemma27B_gold. For both models, training on incorrect teacher-generated data dramatically outperforms training on human-written gold solutions. This strongly supports our hypothesis that the model is able to extract the underlying cognitive artifacts and stitch reasoning traces from the process of the CoT, regardless of the final outcome's correctness. The self-distillation results further corroborate this - training on a model's own incorrect traces (self_distill_wrong) still yields substantial gains over the baseline, demonstrating that even self-generated imperfect data contains a valuable learning signal. The stark difference between the performance of SFT on human_gold data compared to model generated synthetic CoTs, be it stronger teacher generated CoTs or self-generated CoTs, is also very significant. This lag in performance on human_gold data has been studied by [172] showing that models prefer responses generated by a LLMs than those generated by humans even when the CoTs are objectively better.

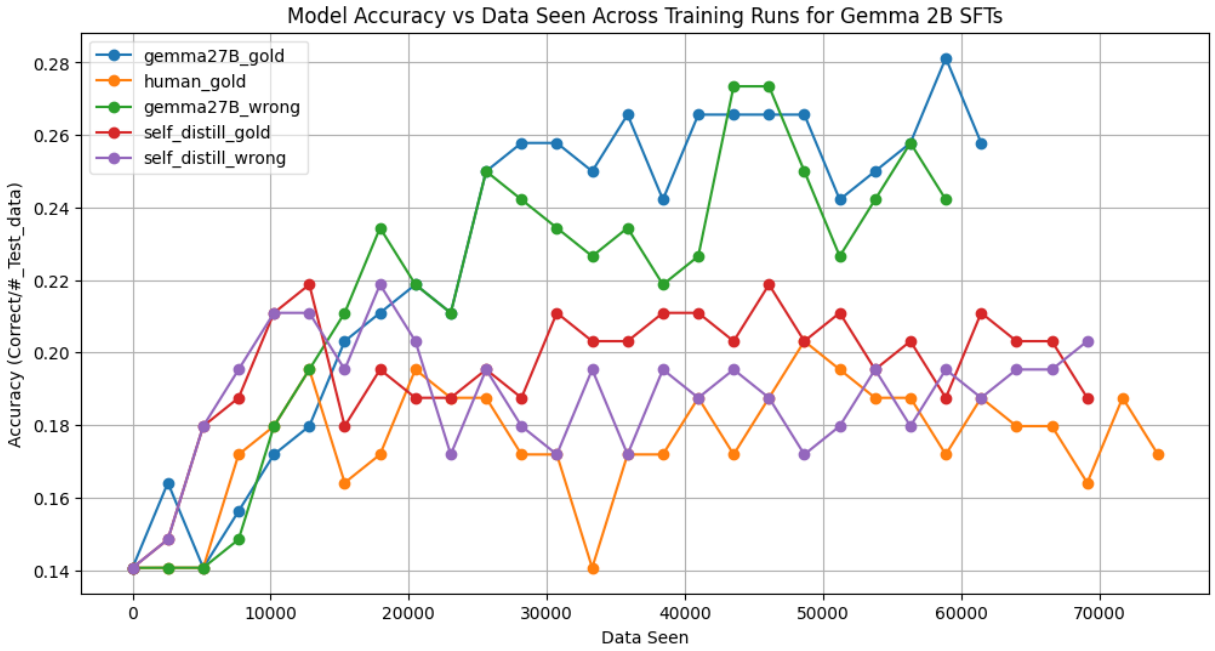


Figure 3.1: Gemma 2-2B SFT results on various datasets. The efficacy of unverified synthetic CoTs generated by a teacher model as well as self-distillation clearly helps learn reasoning skills even better than human_gold dataset.

This opens up the opportunity to scale and leverage huge amounts of synthetic data to bootstrap multiple types of skill acquisition removing the bottleneck of gathering human data which is expensive and often impossible in many domains. It can be clearly attributed to the fact that LLMs are inherently more familiar with other LLM generated response distributions than human responses as evidenced by the lower perplexity scores before SFT. This is further reinforced by the preference of Gemma 2-2B model towards CoTs generated from Gemma 2-27B as compared to that of Qwen 2.5-14B and vice versa in case of Qwen 2.5-1.5B SFT runs.

Analysis of CoT Quality To understand the mechanism enabling this learning from unverified noisy data, we performed a manual analysis on a sample of 100 CoTs generated by the Gemma-27B teacher model. This analysis aimed to categorize the nature of the reasoning in both correct-outcome and incorrect-outcome traces.

The analysis in Table 3.2 provides the crucial ground truth for our framework. It reveals that 68% of the CoTs in the gemma27B_wrong dataset do not suffer from fundamental logical flaws. Instead, their errors are minor and local—calculation mistakes or skipped steps—which correspond directly to the concept of static noise [76]. These traces preserve

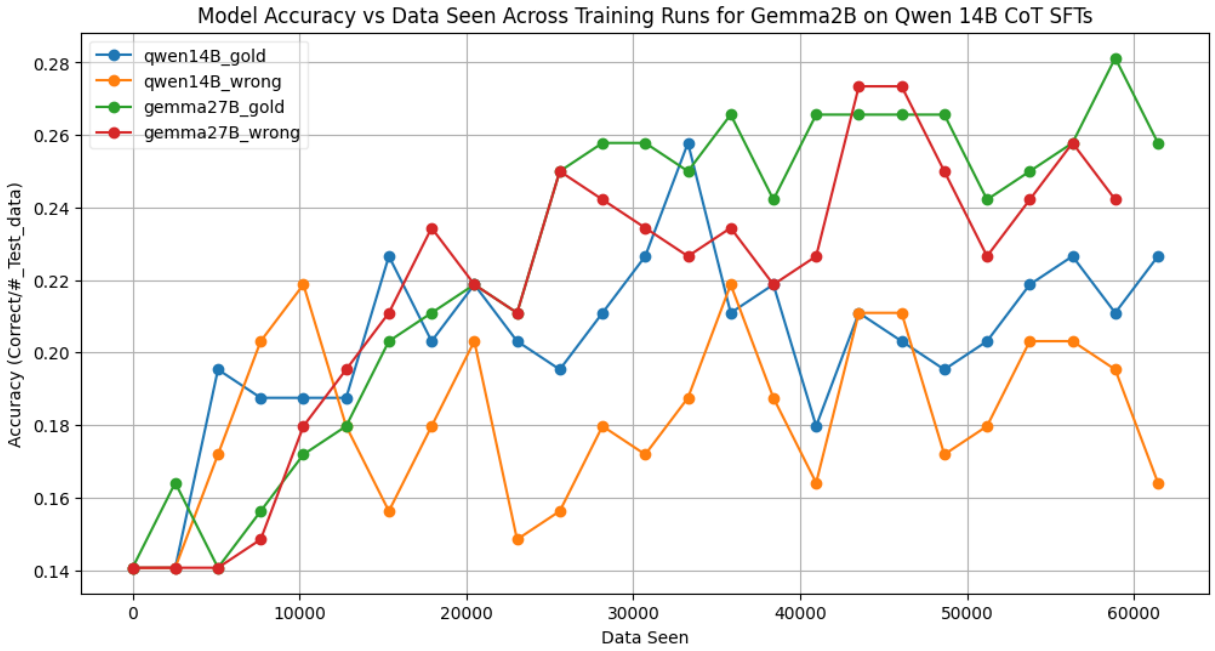


Figure 3.2: Gemma 2-2B SFT results on Gemma2-27B CoTs compared to Qwen2.5-14B CoTs shows the base model's preference towards similar model family generated data.

Table 3.2: Manual analysis of the quality of 100 CoTs generated by the Gemma-27B teacher model. This analysis reveals that a majority of traces leading to an incorrect final answer still contain a largely correct reasoning structure which forms the basis of our proposed method and framework Reg-ReBoot.

CoT Type	**Analysis Category**	**Proportion**
Correct Answer	Fully Correct CoT	71%
	Wrong sub-CoT (calc/approach error)	29%
Incorrect Answer	Minor Problems (calc error, skipped steps)	68%
	Incorrect Reasoning(major logical flaw)	32%

the high-level structure and approach of a correct solution. It is this preserved structure that provides the consistent, learnable signal that the student model bootstraps from, while the varied, local errors are treated as noise and filtered out during the learning process.

The Curious Case of the Qwen Base Model The Qwen 2.5-1.5B base model despite several ablations and hyperparameter sweeps seem to not improve significantly after distillation based SFT. One reasoning being the base model's already strong performance comparable to the teacher models. Furthermore, the baseline performance of Qwen 2.5-1.5B

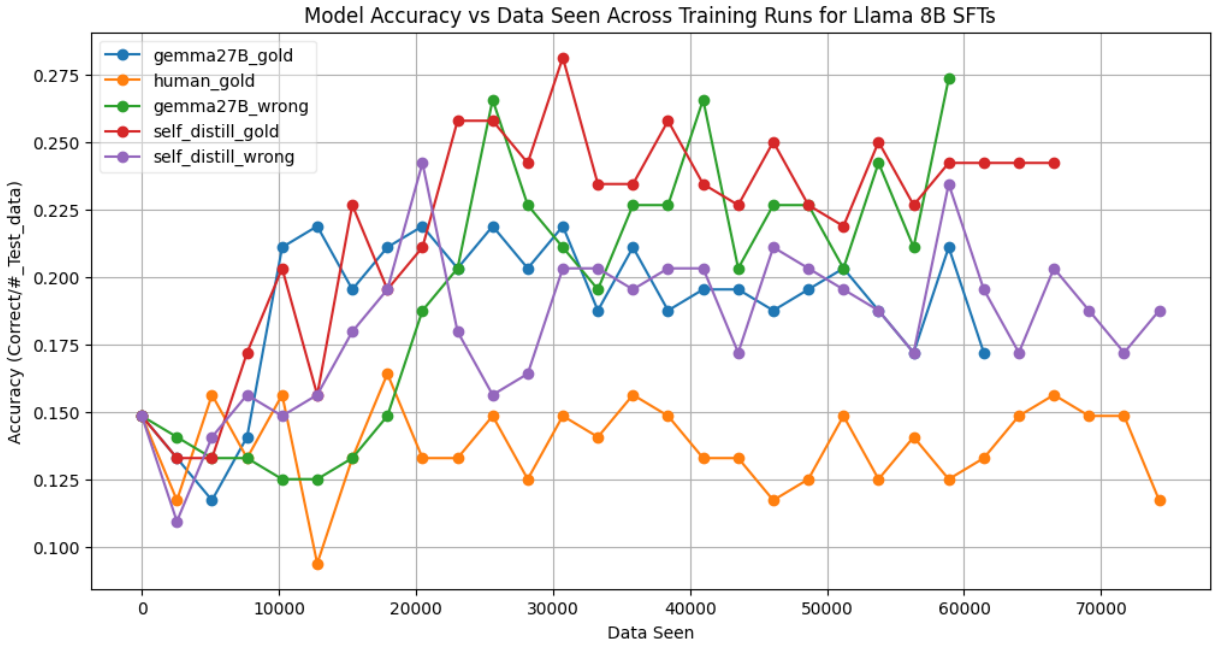


Figure 3.3: Results of SFT on Llama 3.1-8B model on various datasets. Reg-ReBoot provides a good understanding of performance gains in Llama similar in trends to the Gemma model. Llama base model can bootstrap reasoning skills effectively from synthetic unverified datasets and performs significantly better than human_gold data.

with our prompt which is 52.34% is significantly higher than the reported 35% in the technical report. And we perform all experiments with 52.34% as our baseline score. A large number of recent papers have studied the properties of the Qwen family of base models which is one of the most common open model used for downstream reasoning finetuning and they seem to be already very well optimized with strong baseline performance and some reasoning mid-training done already, thereby making them more amenable for direct RL finetuning [222, 180]. While other non-reasoning base models families require the first step of SFT to warm start the acquisition of reasoning abilities which then makes them more suitable for downstream RL finetuning. Despite this, performance on human_gold, qwen14B_gold, and qwen14B_wrong slightly outperform the baseline. Here as well, we can see the model preferring data generated from same model family over others.

Reg-ReBoot demonstrates practical significance - that large-scale imperfect data can substitute for scarce and costly verified solutions, lowering the barrier to scaling reasoning capabilities. It reframes reasoning improvement as learning from noisy but structured patterns rather than requiring curated demonstrations. Practically, this opens a path to bootstrapping reasoning in domains where ground-truth step-by-step solutions are

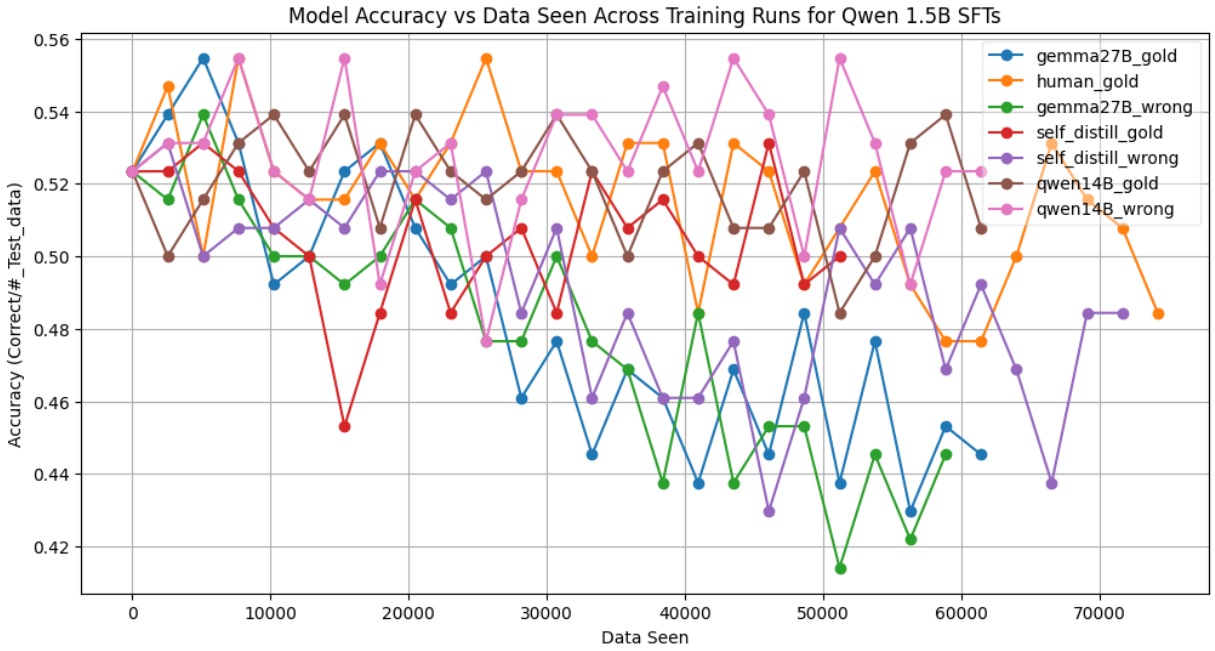


Figure 3.4: Results of SFT on Qwen 2.5-1.5B model on various datasets. AS we can see the gains in performance are moderate compared to the other two model families given the strong base performance of Qwen on Math reasoning already. Still SFT on human_gold, and Qwen 14B distilled CoTs show some gains while the others degrade performance upon scaling data.

unavailable and verification is hard, making reasoning enhancement more cost-effective and generalizable.

3.5 Ongoing Extensions of Experiments and Results

We are working to further analyze this phenomenon and build on the Reg-ReBoot framework to scaling LLM finetuning for reasoning. Two ongoing parts of the work include primarily scaling synthetic data beyond the MATH training dataset's question prompts to include more synthetic question for more diverse and broader coverage and carefully studying the effect of dynamic and static noise to understand the limits of this learning paradigm.

Systematic Analysis of Errors and Noise in the Reasoning CoTs We are performing a controlled experiment to systematically vary the properties of the noisy training data. By precisely manipulating the ratio of static to dynamic errors and the overall noise level, we can quantitatively map the boundaries of the Reg-ReBoot effect and identify the breaking point at which the regularized learning signal loses to the errors. To this end we are

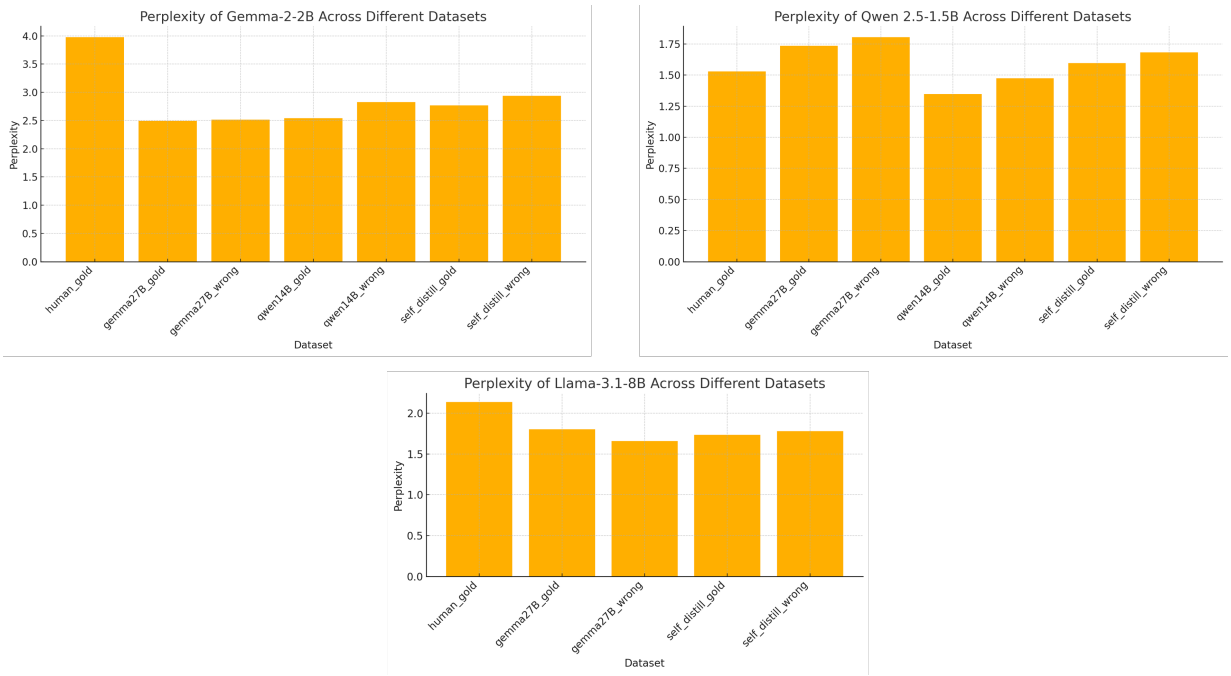


Figure 3.5: Models often prefer synthetic LLM generated data more than gold standard human data measured in terms of perplexity (lower the better).

re-structure the human gold MATH training data and its solution CoTs into more amenable styles and formatting that the base models prefer and then will introduce errors gradually in the gold human dataset which has all sub-parts of the CoTs as correct. This will reveal the resilience of the base models to learn helpful reasoning artifacts as a function of amount and severity of errors introduced (the hyperparameter λ in $P_{\text{mix}}(y|x) = \lambda P_c(y|x) + (1-\lambda)P_e(y|x)$) enabling us to understand the limits of our framework and also the effects of overfitting and memorization of wrong CoTs in the base models.

Datasets with synthetic question prompts as well as unverified synthetic solutions: Given the success of using CoTs without verification, we could leverage another axis to scale the performance of the base LLM models on reasoning - by scaling the number and diversity of problems to learn from and solutions generated by a stronger teacher model which is Gemma 2-27B in our case. We use the generated maths questions prompts derived from the original MATH training set from the works of Google DeepMind Montreal with permission from the teams and authors to open-source these synthetic question dataset and our generated Gemma 2-27B generated synthetic responses for use by the broader community [253]. This synthetic dataset has a total of 7 million plus problem prompts and after filtering for deduplication and high semantic similarity we arrived at a total of 1.2 million question prompts. We then used Gemma 2-27B to generate 16 solutions per prompt and further did a round of filtering for answer format mismatch and token length of solutions to finally reach a dataset of 131 thousand synthetic questions for which we also used Gemma 2-27B to generate their synthetic solution CoTs (which we call gemma27B_syn). We perform no answer verification related filtering on this dataset as we lacked gold truth answers for these synthetic questions anyway. This let us to further study the effect of learning from synthetic questions and their corresponding unverified solution CoTs and also the effect of simply scaling data size in reasoning skill acquisition.

This also enables a simple extension of SFT knowledge distillation setup into a Two-Stage SFT setup as follows:

Algorithm 4: Two-Stage Finetuning

Input: Teacher model P_T, initial student model $P_S(\theta_0)$, unlabeled synthetic problem and solution CoTs set $\mathcal{X}_{\text{syn}}$, high-quality dataset human generated problem set with solution CoTs $\mathcal{D}_{HQ}$

Stage 1: Off-Policy Reasoning Distillation on Noisy Synthetic Problem Prompts;

Generate large, noisy dataset $\mathcal{D}_{T,\text{noisy}}$ using P_T on $\mathcal{X}_{\text{syn}}$ (e.g., Gemma27B_syn);

Fine-tune $P_S(\theta_0)$ on $\mathcal{D}_{T,\text{noisy}}$ to obtain mid-trained model parameters θ_1;

Stage 2: Fine-tuning on High-Quality Human Generated Problem Prompts;

Fine-tune $P_S(\theta_1)$ on $\mathcal{D}_{HQ}$ (e.g., Human Gold or Gemma Gold) to obtain final parameters θ_2;

return $P_S(\theta_2)$;

We are currently working on this direction to further improve the base LLMs reasoning abilities following this two-stage SFT setup trying to analyze the effect of scaling unverified synthetic reasoning data as both in terms of synthetic problems and solution CoTs.

3.6 Conclusion and Future Work

In this work, we challenged the prevailing notion that improving LLM reasoning necessitates massive high quality verified datasets. We introduced Regularized Reasoning Bootstrapping (Reg-ReBoot), a framework grounded in principles of human language acquisition, which posits that LLMs can effectively learn robust reasoning skills from large-scale, unverified, and even incorrect Chain-of-Thought demonstrations.

Our analysis provides a formal mechanism for this phenomenon, connecting the standard supervised fine-tuning objective to the concept of regularization when learning from noisy data. We have shown how the model's inherent inductive bias towards simpler patterns, combined with the statistical properties of math reasoning data, allows it to amplify the consistent signal of correct reasoning primitives while averaging out the diverse noise of idiosyncratic errors.

Our empirical results provide validation for our framework. Through knowledge distillation and self-distillation experiments, we demonstrated that fine-tuning on datasets of CoTs with incorrect final answers yields performance on mathematical reasoning benchmarks that is comparable to training on smaller, human-curated gold datasets. A manual analysis confirmed that these wrong CoTs are not random noise but are rich in partially

correct reasoning structures, providing the necessary signal for the model to bootstrap its capabilities.

The broader implication of this research is a potential efficient alternative towards unsupervised skill acquisition in LLMs, bypassing the predominant need of verifiers and filtered datasets needed to hill-climb in many current LLM tasks. By demonstrating the utility of imperfect data, our work opens a path toward more scalable, cost-effective, and accessible methods for training capable models. Such approaches can allows us to tap into the vast and largely under-utilized troves of unverified real-world data as a valuable resource for teaching machines to think.

Looking forward, some interesting research directions can be built on our existing work:

- Cross-Domain Transfer: The principles of Reg-ReBoot are not inherently limited to math reasoning. A critical next step is to understand which domains of data contain such latent compositional artifacts making them suitable for unsupervised skill acquisition from unverified data using Reg-ReBoot like frameworks. Applying this framework to other complex reasoning domains where verified data is even scarcer and more costly can be an interesting direction to pursue.

- Extensions to Long-CoT Data: Our work currently focuses on short reasoning CoTs. But with the recent rise of R1 distilled open-source long CoT reasoning models along with diverse Long-CoT reasoning datasets can be further utilized to perform better SFT distillation alowing another axis to scale performance with respect to CoT length.

- Integration with Reinforcement Learning: Our findings suggest a powerful two-stage training architecture. Reg-ReBoot can be employed in a first, highly scalable stage to instill foundational reasoning skills using massive amounts of unverified data but as shown by previous works SFT for reasoning can make models capable but not super-human. For this a more targeted stage of RL finetuning can be applied. This hybrid approach, as hinted in our preliminary two-stage SFT plans, could combine the data efficiency and scalability of our method with the and further performance gains afforded by reinforcement learning to build highly capable reasoning systems.

- Scaling the second stage finetuning with Generative Reward Models (GenRMs): With the advent of LLMs trained as next-token predictors for reward modeling, we can continue scaling the second stage of SFT or RL finetuning without requiring rule-based verifiers or gold answers on larger unverifiable problem sets using these GenRMs and also scale inference time compute by simply generating multiple responses verified and ranked using these GenRMs.

Chapter 4

Automating Evaluations for Scalable Oversight and Improving Verifications for Reasoning

4.1 Introduction

The rapid production of content by LLMs and VLMs poses a challenge to traditional human-centric evaluation methods and conventional automatic metrics. Metrics like BLEU [157], ROUGE [127], and METEOR [17] for textual evaluation and CLIPScore [83] for image to text evaluation, often misalign with human judgment and face limitations in assessing creative or nuanced responses. Recent studies suggest using LLMs as novel, reference-independent evaluators by assessing text quality based on predicted sequence likelihoods, bypassing the need for direct reference comparisons [38]. This has motivated researchers [133, 40] to work on improving the evaluation capability of individual LLMs on text evaluation. [258] highlight that large models align more closely with human perceptual processes, thereby enhancing the evaluation of multimedia quality. Consequently, [39] leverage vision language models to provide explainable image quality evaluation by generating textual explanations, assessing fidelity, alignment, and aesthetics.

Surprisingly, despite the potential for improved performance by using ensembles of multiple vision-language models or large language models, there has been limited research on how to align evaluations from multiple VLMs or LLMs with human judgments. The recent advances in inference time compute methods have shown a promising direction to leverage additional compute generating more responses from the same or multiple models to

improve performance in generative tasks [219, 241]. The prospect of thus taking inspiration from these developments and using multiple VLMs or LLMs together in a society of minds [143] setup to solve the complex and nuanced problem of automatic and scalable evaluations is very promising. But it also introduces several uncertainties, including how to select the models, how many models to use, how to manage communication between different models and what prompting structure should be used to maximize the collective effect.

In this paper, we introduce Reason-Feedback-Review (ReFeR), a multi-agent framework drawing inspiration from the academic peer review process to enhance the evaluation of multimodal generative outputs like text generated by an LLM, an image generated by any model, or caption of an image generated by a VLM. By using multiple LLMs or VLMs as evaluators and feedback providers in a system akin to academic peer review, ReFeR enables a comprehensive evaluation of generative outputs across various domains, promoting model self-improvement, reasoning behind evaluation, and consistent score across runs. The paper outlines ReFeR's methodology, including its new prompting schema and the strategic use of LLMs or VLMs in roles parallel to peer reviewers and area chairs, facilitating a multi-dimensional evaluation through a hierarchical framework consisting of two levels: evaluation at the peer level and evaluation at the area chair level. ReFeR thus leverages increased computation at inference time by coordinating multiple models during evaluation to improve performance and by having models themselves produce feedback and final assessment, we introduce an automated model-based evaluation pipeline, reducing the dependence on human evaluators, thereby helping towards scalable oversight of generative model contents [26, 105].

The framework is tested across two NLG evaluation and two multimodal evaluation tasks. Interestingly, the ReFeR framework is generic enough to exhibit strong transfer to reasoning benchmarks as well. The multi-agent hierarchical setup, where multiple peer models deliberate, overseen by the area-chair model, promotes structured reasoning and reflective feedback, enabling complex reasoning and decision-making: we also test its reasoning ability on four reasoning benchmarks. This deliberative process also serves as a scalable generator of high-quality rich feedback-annotated synthetic instruction-tuning datasets. This imbues smaller models with evaluative capabilities more aligned with human preferences. This enables a scalable method to distill the automatic evaluation capabilities of ReFeR-like frameworks into smaller efficient models by fine-tuning them on these datasets, to achieve better correlation with human evaluation.

We present two variants of our proposed framework, ReFeR-Pro and ReFeR-Lite. ReFeR-Lite is $\sim$14$\times$ more test-time compute efficient than ReFeR-Pro. Both the variants outperform strong baselines on both text evaluation datasets: TopicalChat [140] and

SummEval [56]. ReFeR also beats baselines like Clipscore [83], ImageReward [233] and others on caption quality and image generation quality evaluation using ICQD [123] and AGIQA [257] datasets respectively by large margins. Lastly, ReFeR also beats single agent methods (zero-shot CoT [112], self correction [92]), and multi-agent methods like multi-agent debate [48] and multi-agent peer review [234] on 3 out of 4 reasoning datasets, clearly outperforming on average while incurring lower costs than baselines.

To summarize, the primary contributions of our research are as follows: (1) Introduction of a general-purpose tuning-free multi-agent hierarchical framework leveraging inference time compute, called ReFeR, with its two variants, ReFeR-Pro and ReFeR-Lite. (2) We develop a novel prompting schema, with a novel evaluation guidelines component, specifically designed to improve the effectiveness of our framework both evaluation and reasoning tasks. (3) Empirical validation of the framework's evaluation and reasoning skills on four benchmarks each. (4) We conduct an in-depth analysis of our multi-agent framework, addressing key questions such as how to select models, how many models to use, and other critical aspects of model interactions.

4.2 Methodology

4.2.1 ReFeR Framework

Evaluating generative outputs in unverifiable domains without a predefined correct answer, such as assessing the quality of a research paper or open-ended responses, presents significant challenges. Inspired by the hierarchical peer review process in academia, we propose the ReFeR framework, which leverages a hierarchy of language models to systematically evaluate generative outputs. The ReFeR framework consists of two main modules, as depicted in Figure 4.1.

1. Peer Review Body

Let G denote the generative output to be evaluated, E_P represent the prompt for the peer and E_{AC} represent the prompt for the area chair. Let $\mathcal{P} = \{P_1, P_2, \ldots, P_K\}$ be a set of K peer agents, where each P_i is a language model acting as a peer reviewer. Each peer agent independently evaluates G according to E_P, producing a comment C_i and a score $S_i \in \mathbb{R}$. This process is formalized as follows.

$$(C_i, S_i) = \text{Evaluate}_{P_i}(G, E_P), \quad \forall i \in \{1, 2, \ldots, K\} \tag{4.1}$$

2. Area chair Evaluation

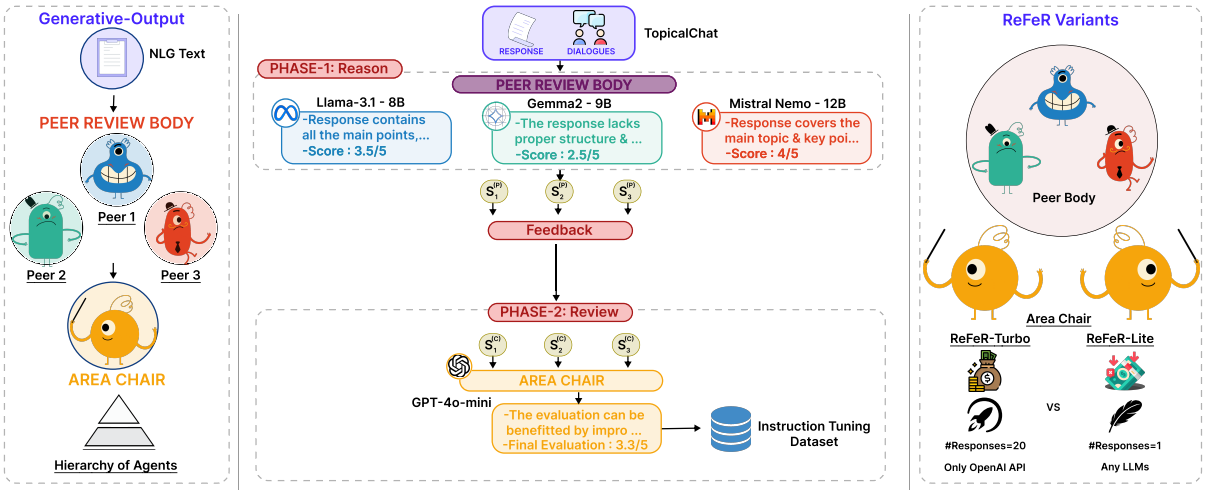


Figure 4.1: Illustration of the ReFeR Framework on the TopicalChat dataset. Refer to Fig. A.2 (in the appendix) for illustration of ReFeR for multimodality and Algorithm 5 showing the framework's working. We use the predictions from AC to create an Instruction tuning dataset which can be used to improve the performance of smaller models as evaluators, shown in Appendix A.10

An area chair agent AC, typically a larger or more capable language model, aggregates and synthesizes the peer reviews, comments and scores to provide the final evaluation. The area chair considers the generative output G, the prompt E_{AC}, and the set of peer reviews $\{(C_i, S_i)\}_{i=1}^{K}$, producing a final comment C_{final} and a final score S_{final}. n is a hyperparameter that denotes the number of responses for a given prompt.

$$\{(C_{\text{AC}}^{(j)}, S_{\text{AC}}^{(j)})\}_{j=1}^{n} = \left\{\text{Evaluate}_{AC}^{(j)}(G, E_{AC}, \{(C_i, S_i)\}_{i=1}^{K})\right\} \tag{4.2}$$

$$S_{\text{final}} = \frac{1}{n}\sum_{j=1}^{n} S_{\text{AC}}^{(j)} \tag{4.3}$$

4.2.2 Prompting Schema

An essential aspect of assessing generative outputs with language model agents involves crafting prompts that elicit high-quality evaluations. Prior work, such as G-Eval by [133], introduced a structured evaluation schema, organizing the prompt into sections: task introduction, evaluation criteria, steps for evaluation, input presentation (context and target), and an evaluation form designed to output a numerical rating only. Subsequently, [40] proposed an adjusted schema named Analyze-Rate, which prioritizes an analytical review followed by scoring, showing improved performance over G-Eval's prompt.

To further refine this approach, we introduce a new module in the evaluation schema called *Evaluation Guidelines* to enhance the model's understanding of the scoring criteria, akin to guidelines provided in traditional academic review processes. Evaluation guidelines can be automatically generated by prompting a language model with the prompt structure and some examples from the dataset and we call this process auto prompt. We give an example of this process in Appendix A.5 showing the *Auto Prompt* for Engagingness prompt for TopicalChat. Alternatively, manually written human annotation guidelines of the dataset can be used. We also modified the evaluation form to include a critical comment or reasoning for the given score. The proposed evaluation schema is shown in Figure A.1.

4.2.3 ReFeR Variants

ReFeR-Pro

ReFeR-Pro leverages a hyperparameter n, representing the number of responses generated by the area chair agent. This variant generates multiple responses ($n = 20$) for each prompt, applying a scoring function that averages the scores across all generated responses, as described in Eq. 4.3.

The final comment C_{final} is the list of all individual comments from the area chair evaluations. While ReFeR-Pro provides superior performance due to generating more evaluations per prompt, it incurs higher computational costs. Additionally, the use of $n = 20$ is often constrained to models from the OpenAI API, as other APIs supporting large models do not support this level of multiple response generation directly. Although it is possible to generate multiple responses by making repeated calls to the model (e.g., running the model 20 times with the same prompt), this approach is computationally expensive and less practical for large-scale evaluation tasks. This usage of the hyperparameter was first suggested by G-Eval and later used by Analyze-Rate.

ReFeR-Lite

To enhance flexibility and reduce computational overhead, we developed ReFeR-Lite, which removes the dependency on the parameter n for the given performance. In this variant, only a single response ($n = 1$) is generated for each prompt, or n is completely removed. This reduction in response generation is reflected in Eq 2, where n is set to 1.

$$(C_{\text{final}}, S_{\text{final}}) = \text{Evaluate}_{AC}(G, E, \{(C_i, S_i)\}_{i=1}^{K}). \tag{4.4}$$

By relying on just one evaluation per prompt, ReFeR-Lite can be used with a wider variety of models, including open-source models, which do not natively support the generation of

multiple responses with a single prompt. Despite generating fewer responses, ReFeR-Lite maintains competitive performance and offers significant cost savings. This makes it a more test-time computing-efficient solution for tasks where computational resources are limited or where evaluating large numbers of samples is required.

Both ReFeR-Pro and ReFeR-Lite use the same peer evaluation structure, but differ primarily in the area chair's response generation and model compatibility. ReFeR-Pro, with $n = 20$, offers potentially higher performance due to generating more evaluations but is restricted to models that support or can simulate multiple response generation with a single prompt. In contrast, ReFeR-Lite provides greater flexibility and cost-efficiency by generating only a single response ($n = 1$) per prompt, making it more suitable for resource-constrained environments.

4.3 Experiments

4.3.1 Datasets

For NLG evaluation, we test our framework on SummEval [56] for summarization evaluation, and TopicalChat [140] for dialogue generation evaluation. For multimodal evaluation, we compare our framework on evaluating two types of task, image-to-text using ICQD (Image Caption Quality Dataset) [123] and text-to-image generation using AGIQA-1k by [257]. For ICQD, we score model-generated captions and compare them with the average human annotated rating for the same. In AGIQA, we assess the quality of AI-generated images in reference to a given prompt and compare it with the mean opinion score (human annotations).

We also test our framework on 4 reasoning datasets: AQuA [128], BBH-DU [193], CSQA [3] and GSM8k [42] which cover various reasoning tasks like Math, Commonsense and Date Understanding. Statistics and details about all the datasets are provided in Table 4.1. For more details about the datasets, refer to Appendix A.7. We test our framework on these reasoning tasks, where our framework answers a reasoning question with the label or numerical Answer after giving the reasoning. We calculate the accuracy of our answers in reference to the gold answers.

4.3.2 Baselines

NLG Evaluation

While the current landscape of models for evaluating NLG responses includes reference-based methods such as BERTScore [255], UniEval [262] and reference-free methods like GPTScore

Table 4.1: Dataset Statistics. We list all the tasks we tackle in our paper and the datasets we used to show results with the number of samples used.For Reasoning, a random subset of 100 was sampled from the original datasets, following [34]. 500 random samples were selected from the original AGIQA-1k to get a well-distributed dataset. We use 864 samples with usable image urls from the ICQD test dataset. We use the full test sets for the NLG Evaluation datasets.

Dataset	Domain	Task	Samples	Answer	Scale
TopicalChat	Dialogue Generation	NLG Evaluation	360	Rating (on 4 metrics)	1-3
SummEval	Summarization	NLG Evaluation	1600	Rating (on 4 metrics)	1-5
ICQD	Image-to-Text	Multimodal Evaluation	864	Caption Score	0-100
AGIQA	Text-to-Image	Multimodal Evaluation	500	Generation Score	0-5
AQuA	Math	Reasoning	100	Option	A-E
CSQA	Commonsense	Reasoning	100	Option	A-E
BBH-DU	Date Understanding	Reasoning	100	Option	A-F
GSM8k	Math	Reasoning	100	Number	-

[62], we do not consider these models as baselines given they were clearly surpassed by G-Eval [133] and later works [40]. Given our work primarily proposes a reference-free LLM-based evaluation for NLG, we do a comparative analysis primarily against G-Eval [133] and Analyze-Rate [40] only.

G-Eval performs evaluation by deploying a single LLM agent. This agent employs Auto-CoT (chain of thought) reasoning and a form-filling paradigm to ascertain the quality of NLG outputs, delivering only scores for the specific dimensions under scrutiny. They use 'n' hyperparameter to generate 20 responses and take the average score.
Analyze-Rate builds upon G-Eval, advocating for an enhanced prompt structure. This methodology incorporates a preliminary analysis phase before scoring, aiming to enrich the evaluative process for NLG tasks. Following G-Eval, they also consider the average score of 20 responses for each sample.

LLM-as-Judge [261] is designed to evaluate LLMs and to rank them potentially creating ChatBOT Arena. We replicated the original setup for TopicalChat dataset.

ChatEval [31] proposed a single model multi-agent framework with varied persona to evaluate various NLG responses on open-ended questions and traditional NLP tasks by leveraging a debating structure among the agents. We reproduced their most optimal setup mentioned in the paper as baseline with 3 roles i.e. 3 agents and 2 discussion turns.

MultiModal Evaluation

For multimodal evaluation, several works like HyperIQA [194], DBCNN [256], IP-IQA [164] were proposed for image quality assessment, but all of these works are deep learning-based methods which leverage and depend on training a capable model. Hence we do not compare our framework against them directly.

CLIP Score [83] evaluates how well an image aligns with a text description by using the CLIP model, which computes similarity scores between images and text embeddings.
Image Reward [233] is a scoring model trained to assess the quality/alignment of generated images with text by comparing them against reference images using a reward model.
Pick Score [107] is another scoring model for the task of image text alignment, which is trained on human preference images 'picked' for a given text.
X-IQE [39] leverages VLMs to evaluate text-to-image generation methods by generating textual explanations. We implement their Alignment dimension experiments to compare with our results on text-to-image generation dataset (AGIQA).

Reasoning

We compare our framework against a variety of baseline methods across different categories. For single-agent methods, we select zero-shot Chain-of-Thought (CoT) and Self-Correct. For multi-agent frameworks, we compare against Multi-Agent Debate and Multi-Agent Peer Review, both of which use a single model acting as multiple agents.

Zero-shot CoT [112] utilizes chain-of-thought prompting to generate reasoning processes and answers using a single agent.
Self-Correct [92] is a single-agent approach that enables an LLM to iteratively evaluate its own outputs, identify errors, and refine its responses through self-reflection.
Multi-Agent Debate [48] involves a group of agents, where each agent observes the solutions provided by others, updates its own solution accordingly, and repeats this process through multiple iterations.
Multi-Agent Peer Review [234] is a multi-agent system in which each agent independently generates a solution, reviews the solutions of others, and assigns confidence scores to its reviews. Agents then revise their initial solutions based on the received peer reviews. This revision is repeated through multiple iterations/rounds of peer review. We used the default number of rounds (3) mentioned by the authors.

4.3.3 Implementation Details

NLG Evaluation: Our framework for NLG evaluation employs Llama-3.1-8B-Instruct [141], Mistral-Nemo-12B [144] and Gemma-2-9B [68] as the peer models and GPT-4o-mini [153] as the area chair model. We use [207]'s API for the peer models, but since these are small open-source models, they can also be deployed locally. For the baselines, we follow the original setups (with GPT-4o-mini) proposed by [133] and [40]. As mentioned in Section 4.2.3, we vary the hyperparameter n for the two ReFeR variants. For more details on other hyperparameters, refer to Appendix A.3.

Multimodal Evaluation: For multimodal evaluation, our framework uses only 2 peers: Gemini-1.5-Flash [46] and GPT-4o-mini [153]. We use GPT-4o [152] as the area chair model. We choose only 2 peers for multimodal evaluation setup considering the cost and availability of VLMs of similar strength. More details on the number of peers and how to choose peers are described in Section 4.5. The baselines like CLIPScore [83], ImageReward [233], PickScore [107] are implemented following the codes provided in their official repositories.

Reasoning: We use the same setup as our NLG evaluation for all our reasoning experiments following similar prompting structure except using evaluation guidelines which is irrelevant in reasoning tasks. All the baselines were implemented and evaluated using the scripts provided by [234] in their official repository.

4.4 Results and Discussions

This section presents the experimental results evaluating ReFeR's effectiveness in assessing text, multimodal outputs, and reasoning capabilities. Experimental details are provided in Section 4.3.3, hyperparameters in Appendix A.3, and prompts in Appendix A.17.

4.4.1 NLG Evaluation

We evaluate ReFeR's performance on two datasets: TopicalChat and SummEval. For TopicalChat, we assess dialog system responses based on four metrics: Coherence, Engagingness, Groundedness, and Naturalness. For SummEval, we evaluate article summaries using Coherence, Consistency, Fluency, and Relevance metrics. Following [133] and [40], we compare the generated scores with human-annotated ground truth using Spearman (ρ) and Kendall-tau (τ) correlations. Results for TopicalChat are presented in Table 4.2, with SummEval results in Appendix A.6. All results are averaged over three runs. The table first shows individual peer performances using our peer prompt, followed by baselines Analyze-Rate [40] and G-Eval [133]. ReFeR Pro outperforms all baselines on most metrics and excels on average. ReFeR Lite, our cost-effective model, ranks second on average despite generating a single response instead of 20 like G-Eval and Analyze-Rate. G-Eval sometimes outperforms Analyze-Rate despite only generating scores, while both Analyze-Rate and ReFeR provide analysis in addition to scores, offering the potential for model improvement. The key findings from this experiment are: (1) Both ReFeR Pro and ReFeR Lite outperform baselines. (2) ReFeR Lite with $n = 1$ also achieves comparable performance while being significantly cheaper. We also provide a statistical test to show the relevance of these automatic evaluations compares to the baselines in Appendix A.13.

While generating multiple responses (e.g., $n = 20$ as in G-Eval) is theoretically possible with any LLM, it poses substantial practical challenges. For instance, evaluating the

TopicalChat dataset (360 samples, 4 metrics) would require approximately 28,800 model calls with an average input token size of 675 for TopicalChat. This approach becomes impractical in terms of cost, time, and computational resources, especially for models without the throughput ($n = 20$) capabilities of the OpenAI API. Hence ReFeR-Lite can be an option in such cases.

Table 4.2: Comparison of ReFeR with baselines for NLG evaluation on the TopicalChat dataset (all baselines and ReFeR uses GPT-4o-mini as the backbone). Results are averaged across 3 runs. The best results are bolded, and the second-best are underlined. *Costs for ReFeR Pro and ReFeR Lite include only AC API cost, as open-source peer models can be deployed locally and so do not involve API costs. Peer model costs based on API pricing from services like [207] are also provided for reference. Relative costs are shown as fractions of the most expensive method. GPT-4o-mini row denotes the performance of the AC model with the peer setup, Peer Average row denotes the correlation when the average of the 3 peer scores is considered.

	Method	**Coherence**		**Engagingness**		**Groundedness**	
		ρ	τ	ρ	τ	ρ	τ
Peer Agents	Llama-3.1-8B	0.380	0.324	0.400	0.342	0.444	0.414
	Mistral Nemo-12B	0.409	0.346	0.594	0.501	0.442	0.414
	Gemma-2-9B	0.536	0.453	0.615	0.527	0.582	0.545
	GPT-4o-mini	0.518	0.438	0.618	0.527	0.589	0.549
	Peer Average	0.547	0.433	<u>0.648</u>	0.519	0.577	0.510
Baselines	Analyze-Rate	0.551	0.465	0.638	<u>0.544</u>	0.615	0.569
	G-Eval	<u>0.581</u>	**0.493**	0.636	**0.546**	0.593	0.555
	LLM-as-Judge	0.510	0.445	0.593	0.519	0.556	0.467
	ChatEval	0.551	0.471	0.624	0.538	0.522	0.428
Ours	ReFeR Pro	**0.592**	0.458	**0.677**	0.536	**0.645**	**0.588**
	ReFeR Lite	0.561	<u>0.479</u>	0.636	0.543	<u>0.618</u>	<u>0.575</u>

	Method	**Naturalness**		**Average**		**Cost**
		ρ	τ	ρ	τ	(Relative)
Peer Agents	Llama-3.1-8B	0.320	0.268	0.386	0.337	0.13
	Mistral Nemo-12B	0.411	0.348	0.464	0.402	0.23
	Gemma-2-9B	0.519	0.430	0.563	0.489	0.20
	GPT-4o-mini	0.540	0.457	0.566	0.493	0.13
	Peer Average	0.512	0.396	0.539	0.447	0.56
Baselines	Analyze-Rate	0.562	<u>0.476</u>	0.591	<u>0.514</u>	0.77
	G-Eval	0.558	0.470	0.592	**0.516**	0.13
	LLM-as-Judge	0.534	0.471	0.548	0.476	0.13
	ChatEval	0.557	**0.478**	0.564	0.479	0.78
Ours	ReFeR Pro	**0.616**	0.473	**0.632**	<u>0.514</u>	1.0*
	ReFeR Lite	<u>0.591</u>	0.416	<u>0.602</u>	0.503	0.13*

4.4.2 Multimodal Evaluation

To assess the multimodal applicability of ReFeR, we conducted experiments on two tasks: image generation quality evaluation using the AGIQA dataset (text-to-image setting) and

image caption evaluation using the ICQD dataset (image-to-text setting). Table 4.3 presents the results of these experiments. Following previous deep learning-based works such as [257], we report Spearman's ρ and Kendall's τ rank correlations. Key findings include the following.

- ICQD dataset: Both variants of ReFeR outperform all baselines. Notably, although individual peers show low correlations, AC effectively countered this, resulting in better correlation.
- AGIQA dataset: ReFeR Pro outperforms all baselines, while ReFeR Lite outperforms ClipScore and X-IQE but falls short of ImageReward and PickScore.

We attribute the performance difference in the AGIQA dataset to the fact that both ImageReward and PickScore involve training based on human preferences, which may have contributed to their superior performance compared to our ReFeR Lite variant. But, our ReFeR-Lite has clearly surpassed a single VLM based method X-IQE by a large margin showing the effectiveness of the framework.

Table 4.3: Multimodal Evaluation Results. Comparison of caption quality and image generation quality score correlations with human scores on ICQD and AGIQA datasets, respectively. *X-IQE is a text-to-image VLM-based method, so we don't show it for Caption Quality.

	Method	**Caption Quality**		**Image Quality**		**Cost**
		ρ	τ	ρ	τ	(Relative)
Peer Agents	Gemini-1.5-Flash	0.135	0.098	0.341	0.268	0.07
	GPT-4o-mini	0.200	0.145	0.502	0.392	0.01
Baselines	Clip Score	0.310	0.233	0.522	0.366	-
	ImageReward	0.433	0.302	<u>0.634</u>	<u>0.451</u>	-
	Pick Score	0.352	0.241	0.627	0.442	-
	X-IQE*	-	-	0.410	0.307	0.05
Ours	ReFeR Pro	**0.497**	**0.347**	**0.657**	**0.467**	1.0
	ReFeR Lite	<u>0.459</u>	<u>0.336</u>	0.599	0.442	0.14

4.4.3 Reasoning

We hypothesize that our framework enhances the overall reasoning capabilities of area chair by utilizing multiple models collaboratively, leading to improved decision-making. To verify this, we compare ReFeR's reasoning capabilities against other frameworks, including zero-shot-CoT, single-agent frameworks, and same-model multi-agent frameworks. Table 4.4 presents the results of these experiments, with all results averaged across 3 runs, following the setup in [34]. Key observations:

- On average, ReFeR outperforms all other baselines across the tested benchmarks.

- In the BBH Date Understanding benchmark, debating-type frameworks like Multi-Agent Debate show better results than ReFeR. This may be attributed to the nature of the benchmark, which involves understanding dates and resolving conflicts. Such tasks benefit from inter-agent discussions, which are possible in a debating setup but not in ReFeR's hierarchical framework.
- ReFeR outperforms baselines on the AQuA benchmark because the hierarchical structure allows the area chair to synthesize peer inputs efficiently, avoiding confusion. In contrast, debate formats may cause models to introduce conflicting reasoning, which is less effective for tasks requiring precise reasoning like AQuA.
- Considering overall cost and performance, both variants of ReFeR demonstrate significant advantages in terms of cost-efficiency compared to corresponding multi-agent models.
- Further analysis on response quality between ReFeR and other methods on GSM8k dataset using reference rationale as premise we get a HHEM score [18] of 0.33 whereas other baselines have a score ranging from 0.102 to 0.115. Higher HHEM score indicates better response quality. More details on the analysis are provided in the Appendix-A.15.

Table 4.4: Experimental results on Reasoning tasks. Comparison of ReFeR performance (accuracy) with single-agent and multi-agent method baselines. All results are averaged across 3 runs. Cost*- Costs are shown as relative to the most expensive method.

Method Type	Methods	AQuA	BBH_DU	CSQA	GSM8k	Average	Cost*
Peer Agents	Llama-3.1-8B	26.3 ± 5.1	28.0 ± 7.8	68.3 ± 4.0	40.0 ± 11.4	40.7 ± 7.1	0.03
	Mistral Nemo-12B	43.0 ± 3.6	55.7 ± 4.6	65.7 ± 6.1	54.7 ± 11.5	54.8 ± 6.5	0.05
	Gemma-2-9B	50.7 ± 2.3	70.3 ± 6.5	75.7 ± 4.5	79.3 ± 4.0	69.0 ± 4.3	0.04
Single Agent	zero-shot-CoT	60.7 ± 1.5	88.0 ± 1.7	76.0 ± 1.7	95.3 ± 1.2	80.0 ± 1.5	0.06
	Self Correction	78.7 ± 0.6	<u>92.7</u> ± 0.6	54.0 ± 1.0	93.3 ± 1.5	79.7 ± 0.9	0.20
Multi-Agent	Multi-Agent Debate	54.0 ± 3.5	**94.3** ± 0.6	<u>79.0</u> ± 1.7	<u>95.7</u> ± 0.6	80.8 ± 1.6	0.97
	Multi-Agent Peer Review	62.0 ± 4.4	91.7 ± 0.6	77.0 ± 1.0	95.3 ± 1.5	81.5 ± 1.9	1.0
Ours	ReFeR Pro	**85.0** ± 1.0	92.0 ± 2.0	**79.3** ± 1.2	**96.0** ± 0.0	**88.1** ± 1.0	0.93
	ReFeR Lite	<u>81.0</u> ± 2.0	91.0 ± 2.0	**79.3** ± 1.2	93.3 ± 2.1	<u>86.2</u> ± 1.8	0.18

4.5 Analysis

In this section, we perform an analysis of the framework to understand the impact of different components and choices.

4.5.1 Peer Ablation

To evaluate the impact of number of peer agents and composition on ReFeR's performance, we conducted a peer ablation study using the TopicalChat dataset shown in Fig. 4.2.

Our findings indicate that increasing the number of peers generally improves the framework's overall correlation, as evidenced by the main branch in Fig. 4.2. We experimented with varying peer combinations and numbers to distinguish between the effects of adding another peer versus a better-performing peer. Due to the impracticality of exploring all possible combinations with six peers, we selected a subset based on individual performances, costs, and model sizes.

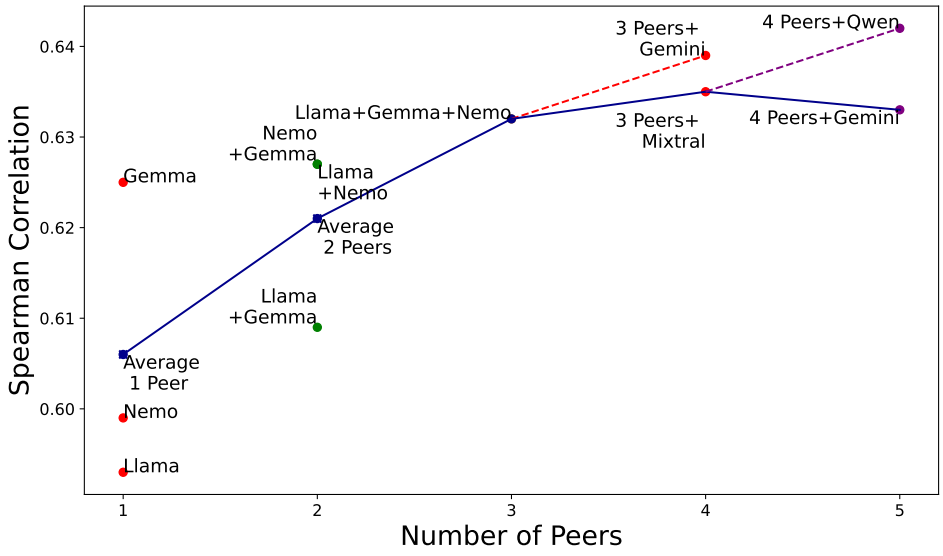


Figure 4.2: Framework Ablation. Results obtained on ReFeR-Pro by progressively adding different peers for the TopicalChat Dataset. The points in the figure indicate the performance of ReFeR when specific labelled peers were used in conjunction with the area chair (GPT-4o-mini). "3 Peers" refers to the Llama, Nemo, and Gemma models being used as peers. "4 Peers" includes the same 3 peers along with the Mixtral model added as fourth peer. Detailed results are presented in Table 4.5.

Fig. 4.2 demonstrates that while the framework's average performance generally increases with more peers, adding a relatively weaker model can result in better performance than the base (1 peer) but not necessarily the highest overall. For instance, with five peers, the combination of four peers plus Qwen yields the best performance, whereas four peers plus Gemini (weaker at this task) performs closer to the three-peer configuration. Notably, the performance gain from four or five peers compared to three peers is not substantial. This observation suggests that using three peers may be an optimal choice, balancing performance improvements with computational efficiency.

4.5.2 Selecting Peers and area chair

For optimal peer selection, we recommend using a group of capable peers chosen based on their individual performances in performance assessment. After assessing individual performances, top-performing peers can be selected considering both cost and performance. As shown in Table 4.5, Gemma2-9B is the top performer across all metrics, while Mistral Nemo 12B and Llama-3.1 8B offer comparable performance at lower costs. Consequently, we selected Gemma2-9B, Llama-3.1 8B, and Mistral Nemo 12B as our peers. This selection is also crucial for enabling local GPU deployment of the peers.

To understand the framework's effectiveness under various conditions, we conducted a study by fixing the peers and changing the area chair. Particularly, we choose an area chair which is relatively weaker than not just GPT-4o-mini but also our best peer, Gemma2-9B, at this task. Hence, we choose Qwen1.5 - 72B. Table 4.6 presents the results using the ReFeR Lite setting on the TopicalChat dataset. We observed that although we used Qwen as AC (whose individual performance is less than the best peer), we get improved performance compared to the respective individual performance. But we see that one of the peer's (Gemma2-9B) has a correlation of 0.568 hence we deduce that if the AC model is relatively stronger than most of the peers then we get improved performance but to get the best results out of the framework we see that we need a larger or better model as AC to better utilize the evaluations done by the peers and incorporate them in it's own evaluation.

This observation aligns with the original analogy of research paper peer review, where the area chair is typically a senior researcher with a potentially better understanding than most peer reviewers, thus being given more importance or final judgment authority. In cases where performance assessment is not feasible to determine the most suitable models, the LLM Leaderboard on [96] can be consulted to select appropriate models based on the specific task requirements, cost considerations, GPU availability, and time constraints.

Table 4.5 shows the individual performances of the 6 open-source models we chose and then the ablation of an increasing number of peers and the ablation of adding different models. The models we used are Llama-3.1-8B, Mistral-Nemo-12B, Gemma-2-9B, Mixtral-8x7B, Gemini-1.5-Pro, Qwen-1.5-72B.

We have also performed AC ablation experiments with another AC model to show the difference of performance based on the model chosen. We have chosen our best peer model Gemma2-9B as the AC model while using the same peer models and their responses, as our original setup. We use the same hyperparameters for the AC model(Gemma) as our original setup. We also show the original GPT-4o-mini results to compare.

From the results of Table 4.7, we can see that the peer performance of Gemma2-9B

Table 4.5: Ablation analysis of the effect of different models used. Comparison of ReFeR Pro results on TopicalChat with Different Peer Configurations. The method column shows what peers were used with the AreaChair (GPT-4o-mini). 4 Peers in the last rows denotes the 3 peers (Llama, Nemo, Gemma) and Mixtral. *These rows show the individual performance of the peers, not the framework's performance when the peer is used.

	Peers Used	**Coherence**		**Engagingness**		**Groundedness**	
		ρ	τ	ρ	τ	ρ	τ
Individual results*	Llama	0.417	0.357	0.418	0.357	0.488	0.455
	Nemo	0.416	0.352	0.567	0.475	0.453	0.424
	Gemma	0.549	0.465	0.623	0.534	0.583	0.545
	Mixtral	0.440	0.373	0.552	0.467	0.491	0.458
	Gemini	0.352	0.300	0.460	0.387	0.498	0.466
	Qwen	0.465	0.399	0.524	0.459	0.471	0.441
	Average	0.440	0.374	0.524	0.446	0.497	0.465
1 Peer	Llama	0.542	0.423	0.603	0.479	0.628	0.556
	Nemo	0.558	0.440	0.684	0.548	0.617	0.555
	Gemma	0.564	0.448	0.680	0.552	0.635	0.578
	Average	0.555	0.437	0.656	0.526	0.626	0.563
2 Peers	Llama+Gemma	0.565	0.440	0.656	0.524	0.593	0.535
	Llama+Nemo	0.577	0.450	0.692	0.553	0.621	0.570
	Nemo+Gemma	0.567	0.443	0.685	0.547	0.622	0.573
	Average	0.570	0.444	0.677	0.541	0.612	0.559
3 Peers	Llama+Gemma+Nemo	0.589	0.458	0.689	0.550	0.623	0.574
4 Peers	3 Peers + Mixtral	0.596	0.463	0.682	0.541	0.629	0.572
	3 Peers + Gemini	0.601	0.469	0.688	0.550	0.644	0.590
5 Peers	4 Peers + Gemini	0.584	0.455	0.686	0.545	0.623	0.572
	4 Peers + Qwen	0.601	0.467	0.682	0.545	0.646	0.592

	Peers Used	**Naturalness**		**Average**	
		ρ	τ	ρ	τ
Individual results*	Llama	0.346	0.289	0.417	0.365
	Nemo	0.396	0.339	0.458	0.397
	Gemma	0.520	0.431	0.568	0.494
	Mixtral	0.469	0.390	0.488	0.422
	Gemini	0.419	0.352	0.432	0.376
	Qwen	0.508	0.437	0.492	0.434
	Average	0.443	0.373	0.476	0.415
1 Peer	Llama	0.599	0.460	0.593	0.479
	Nemo	0.539	0.414	0.599	0.489
	Gemma	0.622	0.481	0.625	0.515
	Average	0.587	0.452	0.606	0.494
2 Peers	Llama+Gemma	0.621	0.481	0.609	0.495
	Llama+Nemo	0.621	0.480	0.627	0.513
	Nemo+Gemma	0.632	0.490	0.627	0.513
	Average	0.624	0.484	0.621	0.507
3 Peers	Llama+Gemma+Nemo	0.626	0.486	0.632	0.517
4 Peers	3 Peers + Mixtral	0.634	0.494	0.635	0.517
	3 Peers + Gemini	0.623	0.485	0.639	0.523
5 Peers	4 Peers + Gemini	0.640	0.495	0.633	0.517
	4 Peers + Qwen	0.637	0.498	0.642	0.526

Table 4.6: Results on TopicalChat using the open-source model Qwen1.5-72B as the area chair. We were unable to include results for ReFeR Pro with Qwen as the area chair due to the limitation of not being able to use n=20.

Method	Coherence		Engagingness		Groundedness	
	ρ	τ	ρ	τ	ρ	τ
(Peer) Llama-3.1-8B	0.417	0.357	0.418	0.357	0.488	0.455
(Peer) Mistral Nemo-12B	0.416	0.352	0.567	0.475	0.453	0.424
(Peer) Gemma-2-9B	**0.549**	**0.465**	**0.623**	**0.534**	0.583	0.545
Qwen (Individual Performance)	0.465	0.399	0.524	0.459	0.471	0.441
ReFeR Lite (Qwen)	0.496	0.422	0.609	0.522	**0.587**	**0.550**

Method	Naturalness		Avg	
	ρ	τ	ρ	τ
(Peer) Llama-3.1-8B	0.346	0.289	0.417	0.365
(Peer) Mistral Nemo-12B	0.396	0.339	0.458	0.397
(Peer) Gemma-2-9B	0.520	0.431	**0.568**	**0.494**
Qwen (Individual Performance)	0.508	0.437	0.492	0.434
ReFeR Lite (Qwen)	**0.527**	**0.450**	0.555	0.486

is relatively lesser than that of GPT-4o-mini. This same trend is observed when using these models as AC in the ReFeR framework. The better performing model GPT-4o-mini gives better correlation as the AC too. This further supports our statements in section 4.5.2, where we show that a stronger model has to be chosen as an AC to get the best performance from the framework.

We also study the effect of using the same model in all our peers as well as the AC in our ReFeR framework and present the results in Apprndix A.16. In short the framework results still surpass the baselines of G-Eval and AnalyzeRate.

4.5.3 Error Analysis

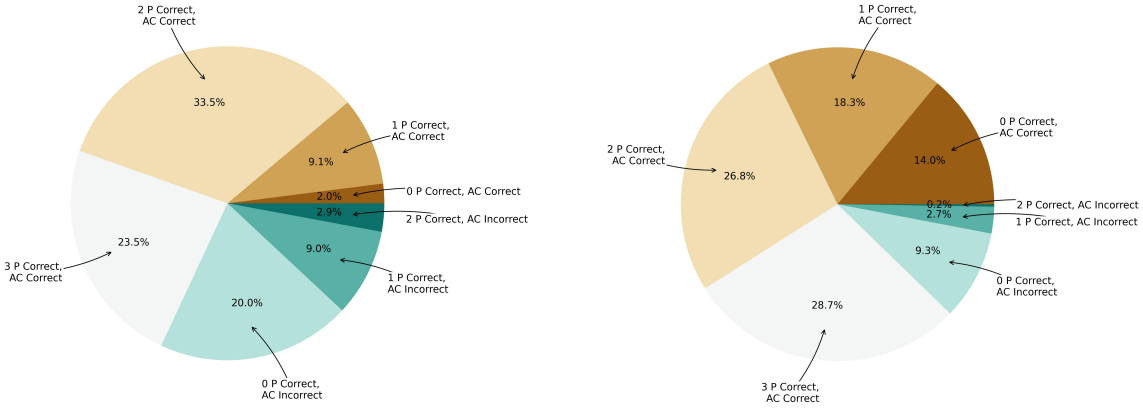
To assess the framework's effectiveness in both evaluation and reasoning tasks, we conducted an error analysis, with results shown in Fig. 4.3. In this analysis, a TopicalChat sample's evaluation score for each metric is considered correct if it falls within a given threshold range of 25%. In the TopicalChat dataset evaluation, the area chair provided correct scores 42.6% of the time when one or two peers provided a correct answer, demonstrating the AC's ability to leverage partially correct peer scores effectively. The AC made mistakes only 11.9% of the time when at least one peer was correct. However, the AC was correct only in 2% of the cases where all the peers were incorrect, suggesting that the AC may require at least one correct peer input to avoid confusion and give a correct score. For reasoning tasks, the AC was incorrect for only 2.9% of cases where atleast one peer is correct, showing similiar observation as evaluation. And the AC was correct 14% of the time, even when all peers were incorrect, indicating a better ability to disregard clearly incorrect answers from the smaller peer models. This suggests that the AC, when using reasoning tasks, may not

Table 4.7: Choosing model for AC: Below Table shows individual performance of AC model and model acting as AC in the ReFeR Framework. The purpose of this table is to understand correlation of capability of the AC and our framework. We See that if gpt-4o-mini is better than gemma in base performance then gpt-4o-mini is better at being AC than gemma

Method	Coherence		Engagingness		Groundedness	
	ρ	τ	ρ	τ	ρ	τ
Gemma-2-9B (Peer Setup)	0.536	0.453	0.615	0.527	0.582	0.545
Gemma-2-9B (n=20)	0.556	0.466	0.617	0.521	0.577	0.538
ReFeR-Pro (Gemma AC)	0.569	0.440	**0.684**	0.540	**0.643**	**0.581**
ReFeR-Lite (Gemma AC)	0.552	0.463	0.624	0.533	0.607	0.567
GPT-4o-mini (Peer Setup)	0.518	0.438	0.618	0.527	0.589	0.549
ReFeR-Pro (gpt-4o-mini as AC)	**0.585**	0.454	0.673	0.535	0.628	0.577
ReFeR-Lite (gpt-4o-mini as AC)	0.552	**0.467**	0.640	**0.550**	0.596	0.558

Method	Naturalness		Average	
	ρ	τ	ρ	τ
Gemma-2-9B (Peer Setup)	0.519	0.430	0.563	0.489
Gemma-2-9B (n=20)	0.530	0.434	0.570	0.490
ReFeR-Pro (Gemma AC)	0.590	0.451	0.621	0.503
ReFeR-Lite (Gemma AC)	0.574	0.484	0.589	0.512
GPT-4o-mini (Peer Setup)	0.540	0.457	0.566	0.493
ReFeR-Pro (gpt-4o-mini as AC)	**0.625**	0.484	**0.628**	0.513
ReFeR-Lite (gpt-4o-mini as AC)	0.599	**0.505**	0.597	**0.520**

always rely on peers and can function independently in such cases.



(a) Evaluation (TopicalChat) **(b)** Reasoning

Figure 4.3: Performance analysis wrt framework scale. Pie-charts showing Peer and AC performance on evaluation and reasoning tasks. (P- Peer model, AC- area chair Model)

More analysis about prompt optimization, communication strategies, inference and test-time compute, and statistical significance tests can be found in Appendix A.4, A.11, A.12 and A.13 respectively.

In terms of practical significance - ReFeR delivers tuning-free, scalable evaluation that is well aligned with human preferences, reducing reliance on human annotators or finetuned evaluators. Beyond single scores, it produces structured feedback that can guide downstream model refinement as automatic evaluators. Its multi-agent structure highlights a general strategy for scalable AI oversight, where diverse evaluators provide checks and balances, making it practical for improving and monitoring generative systems in research and deployment.

4.6 Related Work

NLG & Multimodal Evaluation: Recent advancements in NLG evaluation include GPTScore [62], which uses generative pre-training models to assess text quality, and G-Eval [133], employing a chain-of-thoughts approach with form-filling methodology. [40] highlighted limitations in G-Eval's automated CoT alignment with human evaluations. TIGERScore [99] offers detailed error analysis using fine-tuned Llama-2, while FusionEval [182] integrates auxiliary evaluators with a primary LLM for scoring. X-Eval [130] introduces a two-stage instruction tuning framework for diverse evaluation dimensions. ChatEval [31] proposes a multi-agent referee system using autonomous debating among agents with different personas to evaluate response quality. While similar to our approach, it primarily relies on debate methodology using the same models under varied personas, whereas our method employs diverse models as peers and area chairs with a richer prompting schema. Mixture of Agents (MoA) [216] is another work which explores the how to harness the collective expertise of multiple LLMs. In multimodal evaluation, CLIP Score [83], Image-Reward [233], and Pick Score [107] assess image-text alignment using pre-trained models. Deep learning methods like HyperIQA [194] and IP-IQA [164] have shown improvements in this domain. Later, X-IQE [39] introduced using VLMs for the task of image-quality assessment.

Reasoning using LLMs: Single-agent methods like Zero-shot CoT [112] have improved language models' reasoning capabilities using Chain-of-Thought prompting. Self-correction [92] mimics human self-reflection to address reasoning errors. In multi-agent frameworks, [48] introduced a same-model approach using peer solutions for individual improvement, while [160] proposed embedding-based communication to optimize reasoning. [234] developed a framework inspired by academic peer review, emphasizing iterative improvement through peer feedback. This differs from our method, which involves an area chair reviewing peer responses without direct inter-peer communication. ReConcile [34] structured a multi-model, multi-agent framework as a round table conference, demonstrating enhanced reasoning through discussion and consensus. We expand on why we did not consider ReConcile

as a baseline in the appendix A.8. [223] proposed selecting the most coherent response from multiple reasoning chains, offering an alternative approach to consensus-building and improving reasoning accuracy. For more recent related works discussion, please refer to appendix section A.9.

4.7 Conclusion

In this work, we propose ReFeR (**Re**ason-**Fe**edback-**R**eview), a hierarchical model framework that utilizes smaller yet capable models as peers and a powerful model as the area chair. The area chair leverages the reasoning and feedback from the peer models to produce a final review for evaluating given images or texts. We demonstrate ReFeR's efficacy across two NLG evaluation tasks, two multimodal evaluation tasks, and four reasoning tasks, outperforming various baselines while maintaining favorable cost-performance trade-offs. We also present two variants: ReFeR-Pro and ReFeR-Lite. Notably, our Lite version achieves performance comparable to other works and to ReFeR-Pro, while being significantly more efficient in terms of compute used.

Our framework, while robust in many aspects, has some limitations. One notable constraint is the potential computational cost when using large models as both peers and area chairs, especially in resource-limited environments. Additionally, the framework currently lacks an interactive discussion phase between peer models, which could further improve collective reasoning. In some scenarios, such as when a weaker model is used as the area chair, the performance may not be optimal. Lastly, while our framework has shown promising results on text and image evaluation tasks, it remains untested in other modalities, which could present unique challenges in scaling and complexity and is an interesting area for future research.

Chapter 5

Conclusion

5.1 Summary

The central premise of this thesis has been that the convergence of large-scale Foundation Models and the principles of sequential decision-making represents a critical path toward more general and autonomous AI systems. We began by identifying two powerful but paradigms with some limitations: FMs, which possess vast, high-level knowledge but lack grounding and true agency; and Reinforcement Learning, which offers a robust framework for goal-directed learning but struggles with data efficiency and scalability. The research presented herein has aimed to bridge this gap, tackling three fundamental challenges that arise at the intersection of these fields: building self-improving generative simulators for scalable embodied planning and grounding generative models in physical reality, scaling reasoning in LLMs beyond rule based verifiers and filtered datasets, and scaling automated evaluations of generative model outputs.

The thesis has three distinct works tackling different problems, each contributing toward the development of self-improving agents that can learn and act with less reliance on explicit rewards and perfect human feedback.

1. In Chapter 2, we tackle the problem of learning generative simulators to scale model-based decision making algorithms. We thus introduce VideoAgent, the first self-improving video policy model that served jointly as both the learned world model (WM) or simulator and action policy to perform robot manipulation. We use a goal-conditioned text-to-video diffusion model with a novel self-conditioned consistency objective for correcting video plans. VideoAgent is trained on action-free videos and

is grounded in the physical world using diverse feedback from VLMs and limited online interactions to collect data guided by the WM in a Dagger style iterative loop, eliminating the need for action labels and complicated reward specifications. The learned WM in turn enabled test-time VLM-guided panning over video plans and self-correction of hallucinations, improving agent trajectories as we scaled refinement iterations and online interactions, enabling us to leverage insights from the Bitter Lesson - of effectively using both learning and search.

2. In Chapter 3, we tackle the problem of bootstrapping reasoning in LLMs from unsupervised unverified CoTs without depending on the prevalent idea of using labeled filtered datasets and verifiers. We propose Reg-ReBoot, a framework to learn reasoning from unverified noisy data inspired by cognitive models of children language acquisition. Challenging the predominant reliance on perfectly curated data, we demonstrated that LLMs can learn robust reasoning skills from unverified and even incorrect data. We framed a theoretical and empirical argument that the standard fine-tuning objective, when applied to noisy Chain-of-Thought examples, allows a model to amplify the consistent, underlying latent reasoning primitives while averaging out idiosyncratic errors. Our experiments and ablations showed that training on imperfect, teacher-generated data can yield reasoning capabilities comparable to, or even exceeding, those achieved with human-verified gold datasets, opening the potential for a more scalable and cost-effective path for bootstrapping reasoning in LLMs.

3. In Chapter 4, we give a shot at tackling the final but very crucial step of scaling and automating evaluations to build reliable and scalable oversight methods. We developed ReFeR - a multi-agent, tuning-free framework inspired by the society of minds approach instantiated in the form of the academic peer review process. By structuring a hierarchy of pretrained models as "peers" and an "area chair", ReFeR leverages increased inference-time compute to produce automated, high-quality evaluations for both textual and multimodal outputs. Our results showed that this collaborative, model-based approach not only correlates highly with human judgment but also generates structured, useful feedback that can be distilled for improving downstream model for automated evaluations. Our work provides a viable path for scalable oversight, a critical component for the safe and responsible development of increasingly powerful generative models.

Collectively, with these three works, I try to demonstrate demonstrate how feedback - whether from the physical world (VideoAgent), from the statistical structure of imperfect

data (Reg-ReBoot), or from a society of other AI models (ReFeR) - can be harnessed to create more capable, grounded, and reliable agents. Together, they push the frontier of AI beyond static, passive learning and toward a future of experiential, interactive, and self-improving systems [184].

5.2 Future Work

The research directions that I explored in the thesis require much more work and good science to build truly reliable and safe autonomous agents that are generally capable. While each chapter concludes with specific limitations and future steps, in this section, I try to outline some broader research directions that can be built upon the unified theme of my thesis.

Deepening the Integration of Learning and Search or Planning: VideoAgent demonstrated the potential of using generative models as policies. A natural next step is to more deeply integrate these generative world models with sophisticated planning algorithms. Future work could explore combining generative video foresight with Monte Carlo Tree Search (MCTS) or other model-based RL techniques, allowing an agent to "imagine" and evaluate many possible futures to select an optimal course of action. This would bridge the gap between single-step refinement and long-horizon, multi-step planning.

Closing the Loop: Self-Improving Systems from AI Feedback: The ultimate dream of many AI researchers is a fully autonomous, self-improving agent. Future work could integrate the components presented here into a single, cohesive system. Imagine an agent that uses a ReFeR-like framework to evaluate its own plans generated by a VideoAgent-like policy. The feedback from this self-evaluation could then be used to generate new training data to bootstrap recursive self-improvement. Creating such a system, which learns, acts, evaluates, and improves in a continuous, autonomous cycle, could enable building exploratory open-ended systems like AI-scientists. Although oversight and safety also becomes extremely important in such cases of iterative improving systems and is a very promising direction of further research.

References

[1] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. https://arxiv.org/abs/2308.08201, 2023. arXiv preprint arXiv:2308.08201.

[2] Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. https://arxiv.org/abs/2503.04697, 2025. arXiv preprint arXiv:2503.04697.

[3] Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. Explanations for CommonsenseQA: New Dataset and Models. 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 3050–3065, Online, August 2021. Association for Computational Linguistics.

[4] Alfred V. Aho and Jeffrey D. Ullman. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ, 1972.

[5] Anurag Ajay, Seungwook Han, Yilun Du, Shuang Li, Abhi Gupta, Tommi Jaakkola, Josh Tenenbaum, Leslie Kaelbling, Akash Srivastava, and Pulkit Agrawal. Compositional foundation models for hierarchical planning. *Advances in Neural Information Processing Systems*, 36, 2024.

[6] American Psychological Association. *Publications Manual*. American Psychological Association, Washington, DC, 1983.

[7] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, December 2005.

[8] Galen Andrew and Jianfeng Gao. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40, 2007.

[9] Devansh Arpit. *Deep neural networks are lazy: on the inductive bias of deep learning*. PhD thesis, Massachusetts Institute of Technology, 2019.

[10] Devansh Arpit, Stanislaw Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, A Kanwal, T Maharaj, A Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. *International Conference on Machine Learning*, pages 233–242, 2017.

[11] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[12] Anne C Austin, Karl D Schuler, Sarah Furlong, and Elissa L Newport. Learning a language from inconsistent input: Regularization in child and adult learners. *Language Learning and Development*, 18(3):249–277, 2022.

[13] N. N. Author. Suppressed for anonymity, 2021.

[14] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. In *Robotics: Science and Systems*, 2022.

[15] Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos. In *Neural Information Processing Systems*, 2022.

[16] Can Balioglu, Martin Gleize, Artyom Kozhevnikov, Ilia Kulikov, Tuan Tran, and Julien Yao. fairseq2. http://github.com/facebookresearch/fairseq2, 2023.

[17] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss, editors, *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[18] Forrest Bao, Miaoran Li, Rogger Luo, and Ofer Mendelevitch. HHEM-2.1-Open, 2024.

[19] Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, Ido Shahaf, Oren Tropp, Ehud Karpas, Ran Zilberstein, Jiaqi Zeng, Soumye Singhal, Alexander Bukharin, Yian Zhang, Tugrul Konuk, Gerald Shen, Ameya Sunil Mahabaleshwarkar, Bilal Kartal, Yoshi Suhara, Olivier Delalleau, Zijia Chen, Zhilin Wang, David Mosallanezhad, Adi Renduchintala, Haifeng Qian, Dima Rekesh, Fei Jia, Somshubra Majumdar, Vahid Noroozi, Wasi Uddin Ahmad, Sean Narenthiran, Aleksander Ficek, Mehrzad Samadi, Jocelyn Huang, Siddhartha Jain, Igor Gitman, Ivan Moshkov, Wei Du, Shubham Toshniwal, George Armstrong, Branislav Kisacanin, Matvei Novikov, Daria Gitman, Evelina Bakhturina, Jane Polak Scowcroft, John Kamalu, Dan Su, Kezhi Kong, Markus Kliegl, Rabeeh Karimi, Ying Lin, Sanjeev Satheesh, Jupinder Parmar, Pritam Gundecha, Brandon Norick, Joseph Jennings, Shrimai Prabhumoye, Syeda Nahida Akter, Mostofa Patwary, Abhinav Khattar, Deepak Narayanan, Roger Waleffe, Jimmy Zhang, Bor-Yiing Su, Guyue Huang, Terry Kong, Parth Chadha, Sahil Jain, Christine Harvey, Elad Segal, Jining Huang, Sergey Kashirsky, Robert McQueen, Izzy Putterman, George Lam, Arun Venkatesan, Sherry Wu, Vinh Nguyen, Manoj Kilaru, Andrew Wang, Anna Warno, Abhilash Somasamudramath, Sandip Bhaskar, Maka Dong, Nave Assaf, Shahar Mor, Omer Ullman Argov, Scot Junkin, Oleksandr Romanenko, Pedro Larroy, Monika Katariya, Marco Rovinelli, Viji Balas, Nicholas Edelman, Anahita Bhiwandiwalla, Muthu Subramaniam, Smita Ithape, Karthik Ramamoorthy, Yuting Wu, Suguna Varshini Velury, Omri Almog, Joyjit Daw, Denys Fridman, Erick Galinkin, Michael Evans, Katherine Luna, Leon Derczynski, Nikki Pope, Eileen Long, Seth Schneider, Guillermo Siman, Tomasz Grzegorzek, Pablo Ribalta, Monika Katariya, Joey Conway, Trisha Saar, Ann Guan, Krzysztof Pawelec, Shyamala Prayaga, Oleksii Kuchaiev, Boris Ginsburg, Oluwatobi Olabiyi, Kari Briski, Jonathan Cohen, Bryan Catanzaro, Jonah Alben, Yonatan Geifman, Eric Chung, and Chris Alexiuk. Llama-nemotron: Efficient reasoning models. https://arxiv.org/abs/2505.00949, 2025. arXiv preprint arXiv:2505.00949.

[20] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[21] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn,

Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

[22] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. *arXiv preprint arXiv:2304.08818*, 2023.

[23] Léonard Blier, Corentin Tallec, and Yann Ollivier. Learning successor states and goal-dependent values: A mathematical viewpoint. *arXiv preprint arXiv:2101.07123*, 2021.

[24] Rishi Bommasani, Drew A. Hudson, and Ehsan Adel. On the opportunities and risks of foundation models. *ArXiv*, 2021.

[25] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[26] Samuel R. Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, et al. Measuring progress on scalable oversight for large language models. *arXiv preprint arXiv:2211.03540*, 2022.

[27] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. 2024. *URL https://openai. com/research/video-generation-models-as-world-simulators*, 3, 2024.

[28] Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. *arXiv preprint arXiv:2402.15391*, 2024.

[29] Qingwen Bu, Jia Zeng, Li Chen, Yanchao Yang, Guyue Zhou, Junchi Yan, Ping Luo, Heming Cui, Yi Ma, and Hongyang Li. Closed-loop visuomotor control with generative expectation for robotic manipulation. *2409.09016*, 2024.

[30] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

[31] Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate, 2023.

[32] Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133, 1981.

[33] Annie S Chen, Suraj Nair, and Chelsea Finn. Learning Generalizable Robotic Reward Functions from "In-The-Wild" Human Videos. In *Robotics: Science and Systems*, 2021.

[34] Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. Reconcile: Round-table conference improves reasoning via consensus among diverse llms, 2024.

[35] Pei Chen, Boran Han, and Shuai Zhang. Comm: Collaborative multi-agent, multi-reasoning-path prompting for complex problem solving, 2024.

[36] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

[37] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do not think that much for 2+3=? on the overthinking of o1-like llms. https://arxiv.org/abs/2412.21187, 2025. arXiv preprint arXiv:2412.21187.

[38] Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. Exploring the use of large language models for reference-free text quality evaluation: An empirical study. In Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi, editors, *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 361–374, Nusa Dua, Bali, November 2023. Association for Computational Linguistics.

[39] Yixiong Chen, Li Liu, and Chris Ding. X-iqe: explainable image quality evaluation for text-to-image generation with visual large language models, 2023.

[40] Cheng-Han Chiang and Hung-yi Lee. A closer look into automatic evaluation using large language models, 2023.

[41] Tianzhe Chu and et al. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2504.00001*, 2025.

[42] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

[43] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

[44] Giannis Daras, Yuval Dagan, Alex Dimakis, and Constantinos Daskalakis. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. *Advances in Neural Information Processing Systems*, 36, 2024.

[45] Giannis Daras, Alexandros G Dimakis, and Constantinos Daskalakis. Consistent diffusion meets tweedie: Training exact ambient diffusion models with noisy data. *arXiv preprint arXiv:2404.10177*, 2024.

[46] Google DeepMind. Gemini 1.5-flash. https://deepmind.google/technologies/gemini/flash/, 2024.

[47] Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, and et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*, 2025.

[48] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate, 2023.

[49] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *ArXiv*, abs/2305.14325, 2023.

[50] Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, et al. Video language planning. *arXiv preprint arXiv:2310.10625*, 2023.

[51] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems*, 36, 2024.

[52] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, 2nd edition, 2000.

[53] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

[54] Sugyeong Eo, Hyeonseok Moon, Evelyn Hayoon Zi, Chanjun Park, and Heuiseok Lim. Debate only when necessary: Adaptive multiagent collaboration for efficient llm reasoning, 2025.

[55] Alejandro Escontrela, Ademi Adeniji, Wilson Yan, Ajay Jain, Xue Bin Peng, Ken Goldberg, Youngwoon Lee, Danijar Hafner, and Pieter Abbeel. Video Prediction Models as Rewards for Reinforcement Learning. *arXiv:2305.14343*, 2023.

[56] Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. Summeval: Re-evaluating summarization evaluation, 2021.

[57] Bin Fang, Shidong Jia, Di Guo, Muhua Xu, Shuhuan Wen, and Fuchun Sun. Survey of Imitation Learning for Robotic Manipulation. *International Journal of Intelligent Robotics and Applications*, 2019.

[58] Reuben Feinman and Brenden M. Lake. The shape bias in humans and neural networks. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, pages 362–367, 2018.

[59] Shaoxiong Feng, Xuancheng Ren, Kan Li, and Xu Sun. Multi-view feature representation for dialogue generation with bidirectional distillation. In *AAAI*, 2021.

[60] Chelsea Finn and Sergey Levine. Deep Visual Foresight for Planning Robot Motion. In *IEEE International Conference on Robotics and Automation*, 2017.

[61] Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, et al. Foundation models in robotics: Applications, challenges, and the future. *The International Journal of Robotics Research*, page 02783649241281508, 2023.

[62] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire, 2023.

[63] Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. Specializing smaller language models towards multi-step reasoning. 2023.

[64] Kanishk Gandhi, Denise Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah D Goodman. Stream of search (sos): Learning to search in language. *arXiv preprint arXiv:2404.03683*, 2024.

[65] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. https://arxiv.org/abs/2104.08821, 2022. arXiv preprint arXiv:2104.08821.

[66] Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies, 2021.

[67] Google. Gemma: Google introduces new state-of-the-art open models, 2024.

[68] Google-Research. Gemma-2-9b. https://huggingface.co/google/gemma-2-9b-it, 2024.

[69] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 1994.

[70] Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In *Proc. Interspeech 2019*, pages 1891–1895, 2019.

[71] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang,

Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De

Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto,

Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models. https://arxiv.org/abs/2407.21783, 2024. arXiv preprint arXiv:2407.21783, Llama 3.1 Community License Agreement.

[72] Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, and et al. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.

[73] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[74] Dan Gusfield. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK, 1997.

[75] Rishav Hada, Varun Gumma, Adrian de Wynter, Harshita Diddee, Mohamed Ahmed, Monojit Choudhury, Kalika Bali, and Sunayana Sitaram. Are large language model-based evaluators the solution to scaling up multilingual evaluation?, 2023.

[76] Alex Havrilla and Maia Iyer. Understanding the effect of noise in llm training data with algorithmic chains of thought, 2024.

[77] Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, et al. Videoscore: Building automatic

metrics to simulate fine-grained human feedback for video generation. *arXiv preprint arXiv:2406.15252*, 2024.

[78] Jonathan Heek, Emiel Hoogeboom, and Tim Salimans. Multistep consistency models. *arXiv preprint arXiv:2403.06807*, 2024.

[79] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[80] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021.

[81] Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend, 2015.

[82] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

[83] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning, 2022.

[84] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[85] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[86] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[87] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

[88] Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In *ACL*, 2023.

[89] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[90] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

[91] Zhiting Hu and Tianmin Shu. Language models, agent models, and world models: The law for machine reasoning and planning, 2023.

[92] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet, 2024.

[93] Yi Huang, Jiancheng Huang, Yifan Liu, Mingfu Yan, Jiaxi Lv, Jianzhuang Liu, Wei Xiong, He Zhang, Shifeng Chen, and Liangliang Cao. Diffusion model-based image editing: A survey. *arXiv preprint arXiv:2402.17525*, 2024.

[94] Carla L Hudson Kam and Elissa L Newport. Regularizing unpredictable variation: The roles of adult and child learners in language formation and change. *Language Learning and Development*, 1(2):151–195, 2005.

[95] Hugging Face. Open r1: A fully open reproduction of deepseek-r1. https://github.com/huggingface/open-r1, January 2025.

[96] HuggingFace. Open llm leaderboard. https://huggingface.co/open-llm-leaderboard, 2024.

[97] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[98] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

[99] Dongfu Jiang, Yishan Li, Ge Zhang, Wenhao Huang, Bill Yuchen Lin, and Wenhu Chen. Tigerscore: Towards building explainable metric for all text generation tasks, 2023.

[100] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

[101] Haresh Karnan, Faraz Torabi, Garrett Warnell, and Peter Stone. Adversarial imitation learning from video using a state observer. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2452–2458. IEEE, 2022.

[102] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

[103] Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925*, 2023.

[104] M. J. Kearns. *Computational Complexity of Machine Learning*. PhD thesis, Department of Computer Science, Harvard University, 1989.

[105] Zachary Kenton, Noah Y. Siegel, Janos Kramar, Jonah Brown-Cohen, Samuel Albanie, Jannis Bulian, Rishabh Agarwal, David Lindner, Yunhao Tang, Noah D. Goodman, and Rohin Shah. On scalable oversight with weak llms judging strong llms. In *Proc. of the 37th Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[106] Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[107] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation, 2023.

[108] Donald Knuth. *The TEXbook*. Addison-Wesley, Reading, Massachusetts, 1986.

[109] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. *arXiv preprint arXiv:2310.08576*, 2023.

[110] Tom Kocmi and Christian Federmann. Large language models are state-of-the-art evaluators of translation quality, 2023.

[111] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *NeurIPS*, 2022.

[112] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023.

[113] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

[114] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv preprint arXiv:1712.05474*, 2022.

[115] Abhinav Kumar, Jaechul Roh, Ali Naseh, Marzena Karpinska, Mohit Iyyer, Amir Houmansadr, and Eugene Bagdasarian. Overthink: Slowdown attacks on reasoning llms. *arXiv e-prints*, pages arXiv–2502, 2025.

[116] Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart J Russell, and Pieter Abbeel. Learning Plannable Representations with Causal InfoGAN. In *Neural Information Processing Systems*, 2018.

[117] Philippe Laban, Lidiya Murakhovs'ka, Caiming Xiong, and Chien-Sheng Wu. Are you sure? challenging llms leads to performance drops in the flipflop experiment, 2024.

[118] Leslie Lamport. *LaTeX — A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994.

[119] P. Langley. Crafting papers on machine learning. In Pat Langley, editor, *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pages 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

[120] Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.

[121] Jangwon Lee and Michael S Ryoo. Learning Robot Activities from First-Person Human Videos Using Convolutional Future Regression. In *CVPRW*, 2017.

[122] Youngwoon Lee, Andrew Szot, Shao-Hua Sun, and Joseph J. Lim. Generalizable Imitation Learning from Observation via Inferring Goal Proximity. In *Neural Information Processing Systems*, 2021.

[123] T. Levinboim, A. Thapliyal, P. Sharma, and R. Soricut. Quality estimation for image captions based on large-scale human evaluations. *arXiv preprint arXiv:1909.03396*, 2019.

[124] Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, and Jianfeng Gao. Multimodal foundation models: From specialists to general-purpose assistants, 2023.

[125] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.

[126] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

[127] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[128] Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. 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 158–167, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[129] Chenxiao Liu and Xiaojun Wan. Codeqa: A question answering dataset for source code comprehension. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2618–2632, 2021.

[130] Minqian Liu, Ying Shen, Zhiyang Xu, Yixin Cao, Eunah Cho, Vaibhav Kumar, Reza Ghanadan, and Lifu Huang. X-eval: Generalizable multi-aspect text evaluation via augmented instruction tuning with auxiliary evaluation aspects, 2023.

[131] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023.

[132] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[133] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment, 2023.

[134] Manikanta Loya, Divya Anand Sinha, and Richard Futrell. Exploring the sensitivity of LLMs' decision-making capabilities: Insights from prompt variations and hyperparameters. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[135] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

[136] Ximing Lu, Seungju Han, David Acuna, Hyunwoo Kim, Jaehun Jung, Shrimai Prabhumoye, Niklas Muennighoff, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, and Yejin Choi. Retro-search: Exploring untaken paths for deeper and efficient reasoning. https://arxiv.org/abs/2504.04383, 2025. arXiv preprint arXiv:2504.04383.

[137] Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. Reasoning models can be effective without thinking. https://arxiv.org/abs/2504.09858, 2025. arXiv preprint arXiv:2504.09858.

[138] Sabariswaran Mani, Sreyas Venkataraman, Abhranil Chandra, Adyan Rizvi, Yash Sirvi, Soumojit Bhattacharya, and Aritra Hazra. Diffclone: Enhanced behaviour cloning in robotics with diffusion-driven policy learning. *arXiv:2401.09243*, 2024.

[139] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205, 2017.

[140] Shikib Mehri and Maxine Eskenazi. Usr: An unsupervised and reference free evaluation metric for dialog generation, 2020.

[141] Meta-AI. Llama-3.1-8b-instruct. https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct, 2024.

[142] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors. *Machine Learning: An Artificial Intelligence Approach, Vol. I.* Tioga, Palo Alto, CA, 1983.

[143] Marvin Minsky. *Society of mind*. Simon and Schuster, 1988.

[144] Mistral-AI. Mistral-nemo-instruct-2407. https://huggingface.co/mistralai/Mistral-Nemo-Instruct-2407, 2024.

[145] T. M. Mitchell. The need for biases in learning generalizations. Technical report, Computer Science Department, Rutgers University, New Brunswick, MA, 1980.

[146] Takamichi Miyata. Zen-iqa: Zero-shot explainable and no-reference image quality assessment with vision language model. *IEEE Access*, 12:70973–70983, 2024.

[147] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. https://arxiv.org/abs/2501.19393, 2025. arXiv preprint arXiv:2501.19393.

[148] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A Universal Visual Representation for Robot Manipulation. In *Conference on Robot Learning*, 2022.

[149] A. Newell and P. S. Rosenbloom. Mechanisms of skill acquisition and the law of practice. In J. R. Anderson, editor, *Cognitive Skills and Their Acquisition*, chapter 1, pages 1–51. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1981.

[150] OpenAI. GPT-3.5 Turbo: Language models by openai. https://platform.openai.com/docs/models/gpt-3-5-turbo, 2023.

[151] OpenAI. Gpt-4 technical report, 2024.

[152] OpenAI. Gpt-4o. https://platform.openai.com/docs/models/gpt-4o, 2024.

[153] OpenAI. Gpt-4o-mini. https://platform.openai.com/docs/models/gpt-4o-mini, 2024.

[154] Achiam J OpenAI, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. 2023. *URL: https://arxiv. org/abs/2303.08774*, 2024.

[155] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter

Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155, 2022.

[156] 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.

[157] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

[158] Jyothish Pari, Nur Muhammad Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The Surprising Effectiveness of Representation Learning for Visual Imitation. In *Robotics: Science and Systems*, 2022.

[159] Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive apis, 2023.

[160] Chau Pham, Boyi Liu, Yingxiang Yang, Zhengyu Chen, Tianyi Liu, Jianbo Yuan, Bryan A. Plummer, Zhaoran Wang, and Hongxia Yang. Let models speak ciphers: Multiagent debate through embeddings, 2023.

[161] Priya Pitre, Naren Ramakrishnan, and Xuan Wang. Toward deliberative ai: Multi-agent llms for real-world reasoning. https://vtechworks.lib.vt.edu/items/449ff085-a180-4b40-9b55-d923d220abef, 2025.

[162] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[163] Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7957–7968, Singapore, December 2023. Association for Computational Linguistics.

[164] Bowen Qu, Haohui Li, and Wei Gao. Bringing textual prompt to ai-generated image quality assessment, 2024.

[165] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[166] 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.

[167] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[168] Mohammad Sadegh Rasooli and Joel R. Tetreault. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733, 2015. version 2.

[169] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[170] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

[171] Chris Reinke and Xavier Alameda-Pineda. Successor feature representations. *arXiv preprint arXiv:2110.15701*, 2021.

[172] Xuan Ren, Biao Wu, and Lingqiao Liu. I learn better if you speak my language: Understanding the superior performance of fine-tuning large language models with llm-generated responses, 2024.

[173] Swarnadeep Saha, Peter Hase, and Mohit Bansal. Can language models teach weaker agents? teacher explanations improve students via personalization, 2023.

[174] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):211–229, 1959.

[175] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[176] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[177] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

[178] Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*, 2024.

[179] Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. Concept2Robot: Learning Manipulation Concepts from Instructions and Human Demonstrations. *IJRR*, 2021.

[180] Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in rlvr, 2025.

[181] Pratyusha Sharma, Deepak Pathak, and Abhinav Gupta. Third-person visual imitation learning via decoupled hierarchical controller. In *Neural Information Processing Systems*, 2019.

[182] Lei Shu, Nevan Wichers, Liangchen Luo, Yun Zhu, Yinxiao Liu, Jindong Chen, and Lei Meng. Fusion-eval: Integrating evaluators with llms, 2024.

[183] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[184] David Silver and Richard S. Sutton. Welcome to the era of experience. Preprint / book chapter, DeepMind, 2025. Preprint of a chapter to appear in *Designing an Intelligence* (MIT Press).

[185] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

[186] Jenny L. Singleton and Elissa L. Newport. When learners surpass their models: The acquisition of american sign language from inconsistent input. *Cognitive Psychology*, 49(4):370–407, 2004.

[187] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

[188] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015.

[189] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[190] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

[191] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[192] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[193] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, and et el. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, 2023.

[194] Shaolin Su, Qingsen Yan, Yu Zhu, Cheng Zhang, Xin Ge, Jinqiu Sun, and Yanning Zhang. Blindly assess image quality in the wild guided by a self-adaptive hyper network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[195] Yuxin Su, Yuntao Bai, Jiaming Yu, Long Ouyang, and et al. Llms as verifiers: Efficient reinforcement learning with generative reward models. *arXiv preprint arXiv:2505.00001*, 2025.

[196] Shao-Hua Sun, Hyeonwoo Noh, Sriram Somasundaram, and Joseph Lim. Neural program synthesis from diverse demonstration videos. In *International Conference on Machine Learning*, 2018.

[197] Yiyou Sun, Georgia Zhou, Hao Wang, Dacheng Li, Nouha Dziri, and Dawn Song. Climbing the ladder of reasoning: What llms can-and still can't-solve after sft? https://arxiv.org/abs/2504.11741, 2025. arXiv preprint arXiv:2504.11741.

[198] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, pages 1057–1063, 2000.

[199] Richard S. Sutton. The bitter lesson, 2019.

[200] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, 1998.

[201] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[202] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

[203] Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*, 2019.

[204] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, and many others. Gemini: A family of highly capable multimodal models, 2023.

[205] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[206] Ye Tian, Baolin Peng, Linfeng Song, Lifeng Jin, Dian Yu, Haitao Mi, and Dong Yu. Toward self-improvement of llms via imagination, searching, and criticizing. *arXiv preprint arXiv:2404.12253*, 2024.

[207] Together-AI. Together-ai. https://www.together.ai/, 2023.

[208] Faraz Torabi, Garrett Warnell, and Peter Stone. Recent Advances in Imitation Learning from Observation. In *IJCAI*, 2019.

[209] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, et al. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*, 2017.

[210] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

[211] Alex Wang, Kyunghyun Cho, and Mike Lewis. Asking and answering questions to evaluate the factual consistency of summaries. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online, July 2020. Association for Computational Linguistics.

[212] Haotian Wang, Xiyuan Du, Weijiang Yu, Qianglong Chen, Kun Zhu, Zheng Chu, Lian Yan, and Yi Guan. Learning to break: Knowledge-enhanced reasoning in multi-agent debate system, 2024.

[213] He Wang, Sören Pirk, Ersin Yumer, Vladimir G Kim, Ozan Sener, Srinath Sridhar, and Leonidas J Guibas. Learning a generative model for multi-step human-object interactions from videos. In *Computer Graphics Forum*, volume 38, pages 367–378. Wiley Online Library, 2019.

[214] Hsiang-Chun Wang, Shang-Fu Chen, and Shao-Hua Sun. Diffusion Model-Augmented Behavioral Cloning. *arXiv:2302.13335*, 2023.

[215] Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. Is chatgpt a good nlg evaluator? a preliminary study, 2023.

[216] Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities, 2024.

[217] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. https://arxiv.org/abs/2002.10957, 2020. arXiv preprint arXiv:2002.10957.

[218] 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 *ICLR*, 2023.

[219] 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 *Proc. of the 11th International Conference on Learning Representations (ICLR)*, 2023.

[220] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024.

[221] Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Thoughts are all over the place: On the underthinking of o1-like llms. https://arxiv.org/abs/2501.18585, 2025. arXiv preprint arXiv:2501.18585.

[222] Zengzhi Wang, Fan Zhou, Xuefeng Li, and Pengfei Liu. Octothinker: Mid-training incentivizes reinforcement learning scaling, 2025.

[223] Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration, 2024.

[224] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

[225] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

[226] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.

[227] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[228] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. Godiva: Generating open-domain videos from natural descriptions. *arXiv preprint arXiv:2104.14806*, 2021.

[229] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis, 2023.

[230] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*, 2024.

[231] Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P Lillicrap, Kenji Kawaguchi, and Michael Shieh. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv preprint arXiv:2405.00451*, 2024.

[232] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8121–8130, 2022.

[233] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation, 2023.

[234] Zhenran Xu, Senbao Shi, Baotian Hu, Jindi Yu, Dongfang Li, Min Zhang, and Yuxiang Wu. Towards reasoning in large language models via multi-agent peer review collaboration, 2023.

[235] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng

Zhou, and Zihan Qiu. Qwen3 technical report. https://arxiv.org/abs/2505.09388, 2025. arXiv preprint arXiv:2505.09388.

[236] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers, 2024.

[237] Mengjiao Yang, Yilun Du, Bo Dai, Dale Schuurmans, Joshua B Tenenbaum, and Pieter Abbeel. Probabilistic adaptation of text-to-video models. *arXiv preprint arXiv:2306.01872*, 2023.

[238] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

[239] Sherry Yang, Ofir Nachum, Yilun Du, Jason Wei, Pieter Abbeel, and Dale Schuurmans. Foundation models for decision making: Problems, methods, and opportunities, 2023.

[240] Sherry Yang, Jacob Walker, Jack Parker-Holder, Yilun Du, Jake Bruce, Andre Barreto, Pieter Abbeel, and Dale Schuurmans. Video as the new language for real-world decision making. *arXiv preprint arXiv:2402.17139*, 2024.

[241] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023.

[242] Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning. https://arxiv.org/abs/2502.03387, 2025. arXiv preprint arXiv:2502.03387.

[243] Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. Distilling system 2 into system 1. *arXiv preprint arXiv:2407.06023*, 2024.

[244] Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. Distilling system 2 into system 1. https://arxiv.org/abs/2407.06023, 2024. arXiv preprint arXiv:2407.06023.

[245] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[246] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, Hayden Shively, Adithya Bellathur, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. *arXiv preprint arXiv:1910.10897*, 2021.

[247] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

[248] Weizhe Yuan, Jane Yu, Song Jiang, Karthik Padthe, Yang Li, Dong Wang, Ilia Kulikov, Kyunghyun Cho, Yuandong Tian, Jason E Weston, and et al. Natural-reasoning: Reasoning in the wild with 2.8 m challenging questions. *arXiv preprint arXiv:2502.13124*, 2025. Creative Commons Attribution Non Commercial 4.0 license.

[249] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? https://arxiv.org/abs/2504.13837, 2025. arXiv preprint arXiv:2504.13837.

[250] Tom Zahavy, Brendan O'Donoghue, Guillaume Desjardins, and Satinder Singh. Reward is enough for convex mdps. *Advances in Neural Information Processing Systems*, 34:25746–25759, 2021.

[251] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

[252] Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. Text-to-image diffusion model in generative ai: A survey. *arXiv preprint arXiv:2303.07909*, 2023.

[253] Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction, 2025.

[254] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022.

[255] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020.

[256] Weixia Zhang, Kede Ma, Jia Yan, Dexiang Deng, and Zhou Wang. Blind image quality assessment using a deep bilinear convolutional neural network. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(1):36–47, January 2020.

[257] Zicheng Zhang, Chunyi Li, Wei Sun, Xiaohong Liu, Xiongkuo Min, and Guangtao Zhai. A perceptual quality assessment exploration for aigc images, 2023.

[258] Zicheng Zhang, Yingjie Zhou, Chunyi Li, Baixuan Zhao, Xiaohong Liu, and Guangtao Zhai. Quality assessment in the era of large models: A survey, 2024.

[259] Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Long is more for alignment: A simple but tough-to-beat baseline for instruction fine-tuning. https://arxiv.org/abs/2402.04833, 2024. arXiv preprint arXiv:2402.04833.

[260] Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. Progressive-hint prompting improves reasoning in large language models. *arXiv preprint arXiv:2304.09797*, 2023.

[261] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

[262] Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. Towards a unified multi-dimensional evaluator for text generation, 2022.

[263] Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. Towards a unified multi-dimensional evaluator for text generation, 2022.

[264] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *ICLR*, 2023.

[265] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers, 2023.

[266] Chuning Zhu, Xinqi Wang, Tyler Han, Simon S Du, and Abhishek Gupta. Distributional successor features enable zero-shot policy optimization. *Advances in Neural Information Processing Systems*, 37:124612–124639, 2024.

[267] Junchen Zhu, Huan Yang, Huiguo He, Wenjing Wang, Zixi Tuo, Wen-Huang Cheng, Lianli Gao, Jingkuan Song, and Jianlong Fu. Moviefactory: Automatic movie creation from text using large generative models for language and images. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9313–9319, 2023.

[268] Xuekai Zhu, Biqing Qi, Kaiyan Zhang, Xingwei Long, and Bowen Zhou. Pad: Program-aided distillation specializes large models in reasoning. *arXiv preprint arXiv:2305.13888*, 2023.

APPENDICES

Appendix A

Appendix and Supplementary Material for Chapter 4: ReFeR

A.1 Future Works

Future research can explore incorporating additional elements from the academic peer review process, such as the author discussion phase, to simulate a more interactive review environment. Expanding the framework to include evaluations beyond text and images, such as video and audio content, could further enhance its applicability. Another promising direction is to develop various communication strategies between peers and the area chair to optimize evaluation and feedback cycles. Moreover, experimenting with different numbers of area chairs of varying strength could help in understanding the impact of multiple, potentially conflicting, judgments on the final evaluation outcomes.

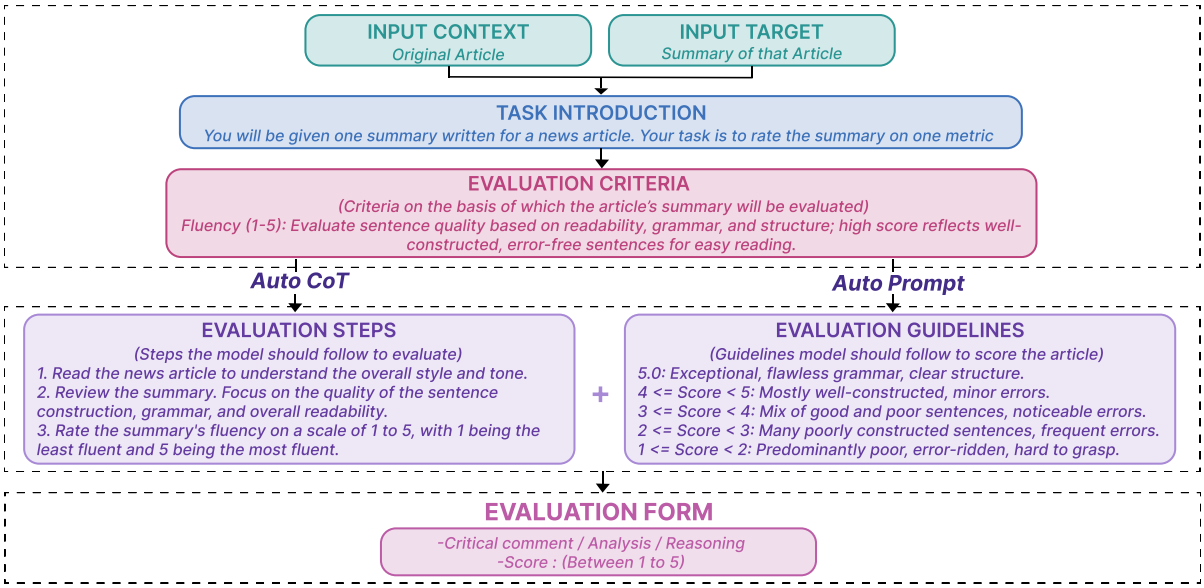


Figure A.1: Prompting Schema

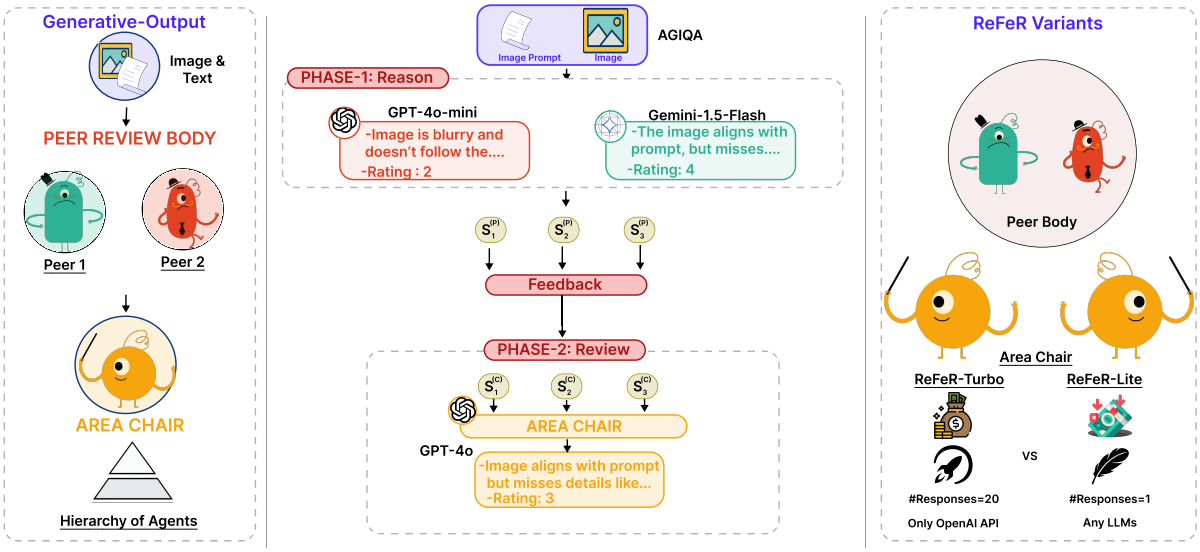


Figure A.2: Illustration of ReFeR for Multimodal evaluation shown on AGIQA dataset. A similar version of ReFeR working on textual TopicalChat dataset is shown in 4.1.

A.2 ReFeR Algorithm

Algorithm 5: ReFeR Framework for Evaluating Generative Outputs

Input :

- Generative output O (text or image)
- Prompting schema E_P for peers and E_{AC} for area chair
- Peer Models $P = \{P_1, P_2, P_3, \ldots, P_K\}$ (K Peers)
- Area Chair Model AC
- Variant $V \in \{\text{ReFeR-Pro}, \text{ReFeR-Lite}\}$
- Number of responses n (only for ReFeR-Pro)

Output :

- Final Evaluation Score S_{final}
- Constructive Feedback C_{final}

Phase 1: Peer Review Body Evaluation;
foreach *peer model* $P_i \in P$ **do**
  `// Each peer independently evaluates` O `using prompting schema` E_P
  $(C_i, S_i) \leftarrow \text{Evaluate}_{P_i}(G, E_P)$;
end
Phase 2: Area Chair Evaluation;
if $V =$ *ReFeR-Lite* **then**
  $n \leftarrow 1$;
end
else
  $n \leftarrow 20$;
end
for $j = 1$ **to** n **do**
  $(C_{\text{AC}}^{(j)}, S_{\text{AC}}^{(j)}) \leftarrow \text{Evaluate}_{AC}^{(j)}(G, E_{AC}, \{(C_i, S_i)\}_{i=1}^{K})$;
end
`// Compute final score`
$S_{\text{final}} \leftarrow \frac{1}{n} \sum_{j=1}^{n} S_{\text{AC}}^{(j)}$;
$C_{\text{final}} \leftarrow$ Choose 1 randomly whose score is closest to $S_{\text{final}}(C_{\text{AC}}^{(1)}, \ldots, C_{\text{AC}}^{(n)})$;
return *Final Evaluation Score* S_{final}*, Constructive Feedback* C_{final};

Mathematical Notation Summary:

- G: Generative output to be evaluated.
- E_P: Prompt of peer.
- E_{AC}: Prompt of area chair.
- P_i: Peer agent i, for $i = 1, \ldots, K$.
- K: Number of peer agents.
- C_i: Comment from peer agent P_i.
- S_i: Score from peer agent P_i.
- AC: area chair agent.
- n: Number of independent evaluations by AC in ReFeR-Pro.
- $C_{\text{AC}}^{(j)}$, $S_{\text{AC}}^{(j)}$: Comment and score from the j-th evaluation by AC.
- C_{final}, S_{final}: Final comment and score.

In summary, the ReFeR framework formalizes the evaluation of generative outputs by modeling the process after the hierarchical peer review system, with mathematical rigor to facilitate clarity and reproducibility. This approach not only enhances the evaluation accuracy but also provides constructive feedback, aligning closely with human judgment and expectations in complex evaluation scenarios.

A.3 Hyperparameters

For the ReFeR NLG Evaluation setup, following Analyze-Rate [40], we set these hyperparameters as follows, for the AreaChair GPT-4o-mini model- *temperature=1, max_tokens=256, top_p=1, frequency_penalty=0, presence_penalty=0, stop=None, n=20* (varies for ReFeR Lite and Pro). For the peer models, we use the default hyperparameters except for the *max_tokens=128*. For multimodal evaluation, we use the same setup for the AC, but for the peers, we increase the *max_tokens* from 128 to 192 tokens. For reasoning tasks, we follow the NLG evaluation setup for the area chair, but we don't set any limit on the *max_tokens* hyperparameter. For the peer models, we increase *max_tokens* to *256* and set the hyper-parameters *temperature=1, top_p=1*.

A.4 Prompt Optimization

Prompt optimization methods utilizing LLMs, such as OPRO [236], APE [265], and ProTeGi [163], often employ text-gradient or feedback-based techniques to refine prompts. These methods typically involve providing a capable LLM with error examples and obtaining feedback, which serves as a text gradient to adjust the prompt for improved performance. Such approaches have demonstrated effectiveness for short questions/requests and relatively simple tasks.

Table A.1 compares our prompt with an optimized prompt using ProTeGi on the TopicalChat dataset. We utilized the default ProTeGi settings with GPT-4o-mini as the

optimizer LLM. Our findings indicate that prompt optimization is time-consuming and incurs higher costs than evaluation itself due to the iterative improvement process of prompt optimization over a test set, and that too with long inputs for complex tasks like this. Moreover, our prompts with the proposed structure yield better correlation than the optimized prompts from ProTeGi.

We attribute this outcome to the limitations of these methods when dealing with extensive inputs, such as conversation history and responses in TopicalChat, where entire dialogues are provided to evaluate and rate NLG output on various metrics. Unlike G-Eval, which only provides scores, methods like Analyze-Rate and ours improve scores based on generated analyses. In these cases, both the analyses and ratings are crucial for understanding errors, as the singular numerical rating value offers insufficient insight into prompt issues. Additionally, even when detailed analyses are provided in multiple error examples for the prompt optimization, the gradient-based approach may struggle with long contexts, making it challenging for the model to identify specific prompt deficiencies and provide useful feedback.

Table A.1: Analysis of Prompt Optimization. Comparison of Average (across 4 metrics) results for different prompts on TopicalChat dataset.

Method	ProTeGi		Ours	
	ρ	τ	ρ	τ
Llama-3.1-8B	0.347	0.303	0.386	0.337
Mistral-Nemo-12B	0.387	0.336	0.464	0.402
Gemma-2-9B	0.511	0.444	0.563	0.489
ReFeR Pro	0.625	0.511	0.628	0.513

The table shows the results of average results across 4 metrics for peers and the framework for Prompt optimized by ProTeGi vs Our Prompt generated through our prompting schema. We can see that the peers' performance declines with the optimized prompts. These prompts were the best prompts after 3 rounds of Prompt-Optimization with ProTeGi. But still their performance falls short to our prompting schema. And even though the framework is relatively close, it would still make a point on how the effort and costs for the prompt optimizations would not be worth it. Running ProTeGi prompt optimizations alone for peers and area chair costs ~ 4 times the cost of evaluating using ReFeR-Pro.

Prompt Ablation

Table A.2 shows the ReFeR framework performance on TopicalChat Dataset with different prompts. We can see from the results that our prompting Schema gives the highest performance with Analyze-rate being the second. From the significant difference, we can see that G-eval scores only prompting doesn't work for our ReFeR framework.

Table A.2: Analysis of Prompt Ablation. Average results (across 4 metrics) on TopicalChat Dataset of ReFeR-Pro with different Prompt structure.

	Prompt Schema	Average	
		ρ	τ
ReFeR-Pro	G-Eval	0.568	0.454
	Analyze-Rate	0.592	0.510
	Ours	0.628	0.513

A.5 Auto Prompt

Example Implementation of Autoprompt:

Input AutoPrompt:

AutoPrompt for Engagingness Evaluation for TopicalChat

You are tasked with creating a clear and concise prompt for a task based on the provided prompt structure and examples from dataset. The prompt should be written in such a way that it can be easily understood and followed by another LLM or human user performing the task. Your prompt should include the following:

- *A brief overview of the task.*
- *Evaluation criteria explaining what metric the evaluation is going to be on.*
- *Clear instructions for how to approach the task or evaluation steps.*
- *Use the examples of dataset, analyze and understand how it is evaluated for the given metric and give a detailed Evaluation guidelines which will tell when to give a particular score.*

Example Prompt Structure:

You will be presented with a conversation between two individuals and given a potential response for the next turn in the conversation, along with a fact that the response is based on.

Your task is to evaluate the response on a single metric: Engagingness. The rating must be given after giving the analysis too.

Evaluation Criteria:

Engagingness (1-3): Assess whether the response is dull, moderately interesting, or highly engaging.

Evaluation Steps:

1. *Read the conversation, the corresponding fact and the response carefully.*
2. *Rate the response on a scale of 1-3 for engagingness, according to the criteria above.*

Please ensure the prompt explains the rating scale from 1 to 3 clearly.

Examples for Task:

Example 1:

Conversation History:
Person A: "I heard that flamingos get their pink color from their diet."
Person B: "Really? I didn't know that! What do they eat to turn pink?"

Fact: Flamingos get their pink color from consuming shrimp and algae rich in carotenoids.

Response: "Yes! They eat shrimp and algae, which contain pigments called carotenoids that give them their pink color."

Rating: 3

Example 2:

Conversation History:
Person A: "I just rewatched Interstellar. The way they depict time dilation near a black hole is incredible."
Person B: "Yeah, the science behind it is pretty solid. What did you think of the emotional storyline with Cooper and his daughter?"
Person A: "I thought it was powerful how they tied in the human element with such a complex scientific concept."
Person B: "Do you think the film's portrayal of black holes was accurate?"

Fact: Interstellar consulted physicist Kip Thorne to ensure the black hole depiction was as scientifically accurate as possible, especially with the time dilation effects.

Response: "It was fine."

Rating: 1

The above prompt is an example of Auto Prompt for the Engagingness metric in TopicalChat dataset. The text marked in blue, i.e., the example prompt structure and the examples are the inputs to be given by the user to get the evaluation guidelines to be used in our proposed prompting schema. In the example above we use the prompt structure from Analyze-Rate engagingness prompt. Hence by prompting a capable LLM with an Auto Prompt crafted for a task giving the prompt structure and examples, we can obtain a prompt following our prompting schema for that task, including evaluation guidelines.

Output based on the above prompt:

Engagingness Peer Prompt

You will be given a conversation between two individuals. You will then be given one potential response for the next turn in the conversation. The response concerns an interesting fact, which will be provided as well.

Your task is to rate the responses on one metric. Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria: Engagingness (1-3): Assess whether the response is dull, moderately interesting, or highly engaging.

Evaluation Steps:

1. *Read the conversation, the corresponding fact and the response carefully.*
2. *Rate the response on a scale of 1-3 for engagingness, according to the criteria above.*

Evaluation Guidelines:
- A score of 1 (dull) means that the response is generic and dull.
- A score of 2 (somewhat interesting) means the response is somewhat interesting and could engage you in the conversation (e.g., an opinion, thought).

- A score of 3 (interesting) means the response is very interesting or presents an interesting fact.

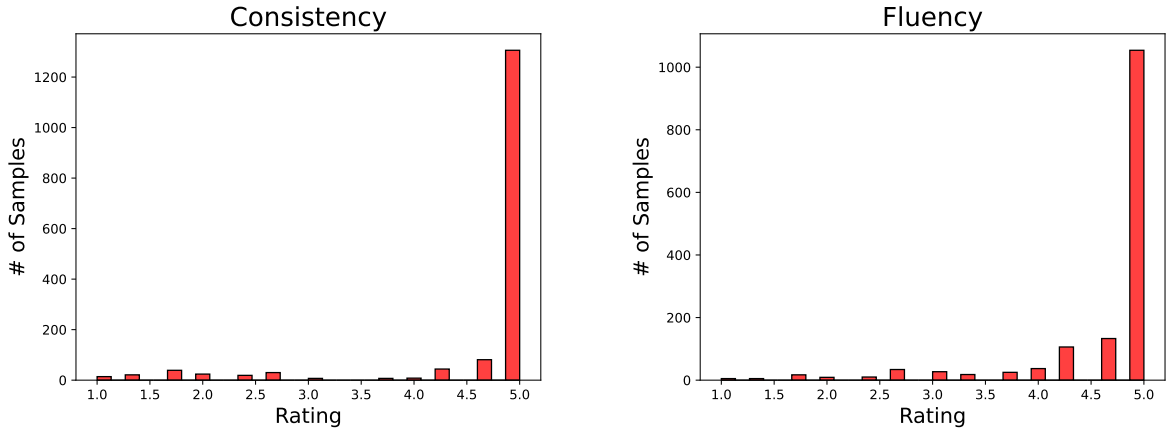
A.6 SummEval Results

We test our framework on the SummEval Dataset, comparing it with the baselines G-Eval and Analyze-Rate. We first show our individual peer performances, then the baselines, and finally, the two variants of our ReFeR framework. Before delving deep into the results of this benchmark, it is important to discuss the dataset distribution of SummEval. As shown in Fig. A.3, the dataset is highly skewed for the consistency and fluency metrics, with almost 1300+ and 1100+ samples having a score of 5 for consistency and fluency, respectively.

Table A.3: Performance analysis on SummEval dataset. Comparison of various methods for NLG evaluation on SummEval.

	Method	Coherence		Consistency		Fluency		Relevance		Average	
		ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ
Peer Agents	Llama-3.1-8B	0.351	0.287	0.425	0.381	0.307	0.277	0.361	0.295	0.361	0.310
	Mistral Nemo-12B	0.367	0.296	0.383	0.340	0.239	0.211	0.368	0.303	0.339	0.287
	Gemma-2-9B	**0.560**	**0.460**	0.474	**0.433**	0.387	**0.347**	0.517	0.422	0.484	**0.415**
Baselines	Analyze-Rate	0.533	0.392	0.382	0.305	0.353	0.283	0.430	0.320	0.425	0.325
	G-Eval	0.509	0.387	0.475	0.386	0.334	0.290	**0.571**	**0.433**	0.472	0.374
Ours	ReFeR	0.528	0.403	**0.478**	0.390	**0.425**	0.342	0.521	0.395	**0.488**	0.382
	ReFeR Lite	0.483	0.400	0.472	0.420	0.360	0.324	0.472	0.397	0.447	0.385

The skewed distribution in the Summeval dataset creates an imbalance in correlation evaluations. Smaller models, such as Gemma-2-9B, which may lack sensitivity to subtle differences in the data (e.g., article and summary), often give uniformly high scores that mimic the skewed human annotations, resulting in higher correlations. However, this does not reflect the model's true ability to understand and follow instructions. In contrast, larger models like GPT-4o-mini, which adhere more strictly to evaluation guidelines, tend to generate more varied scores. This often leads to lower correlations due to the skewed nature of the human annotations. Additionally, in cases where the ReFeR framework provides consistent scores, the result is a high number of tied predictions, which further lowers Kendall's tau coefficient due to the large number of tied pairs. This complicates the interpretation of performance for larger models and more advanced frameworks, as the lack of variability in the dataset hampers an accurate assessment of model effectiveness. Given these challenges, although SummEval is a popular benchmark dataset used for NLG evaluation, we do not consider Summeval to be an appropriate benchmark for testing our



(a) Consistency **(b)** Fluency

Figure A.3: Ratings Distribution. We show the distribution of human annotations for Consistency, FLuency metrics in the SummEval dataset

methods, unless a uniformly distributed sample can be extracted—a difficult task given the inherent skewness of the annotations. Hence, even though ReFeR-Pro outperforms other baselines on this dataset, we present these results only in the appendix.

A.7 Datasets

NLG Evaluation

- **SummEval** [56] provides human assessments on four critical dimensions of summarization quality: Coherence, Consistency, Fluency and Relevance, utilizing the CNN/DailyMail dataset [81] as its foundation.
- **TopicalChat** [70] is a dataset of conversations. We use the dataset created by [140] using the TopicalChat dataset in which they give a possible next response generated by a language model for a given conversation history, and the human annotation score of the response on five attributes: Coherence, Engagingness, Groundedness, Naturalness, and Understandability. We exclude Understandibility, following the previous works G-Eval, Analyze-Rate and Uni-Eval[1]

[1]Uni-Eval shows results on the 4 metrics and uses the Understandability metric for transfer experiment, hence only 4 dimensons are shown in the following works. Refer to [262] for more details.

MultiModal Evaluation

- **ICQD** (Image Caption Quality Dataset) [123] focuses on the task of Quality Estimation (QE) for image captions. We use the test dataset which provides human ratings (0/1) on quality. We scale these average ratings to a scale of 0-100 for our evaluation.
- **AGIQA** (AI Generated Image Quality Assessment) [257] presents a AGI quality assessment database, AGIQA-1K, which consists of 1,080 AGIs generated from diffusion models. They provide MOS (Mean Opinion Score) in the range of 0-5. We have observed that the dataset is skewed around certain scores around 3-3.5. So to test on a subset which has variance of image quality ratings, we select 500 samples, such that the data more or less equally spread on the rating range (0-5).

Reasoning

- **AQuA** (Algebra Question Answering) [128] dataset is designed to assess a model's reasoning abilities in solving algebraic word problems. It consists of multiple-choice math questions, where the model must understand and compute the correct answer from several options.
- **BBH-DU** (Big Bench Hard Date Understanding) [193] dataset is part of the BIG-Bench benchmark. It focuses on testing a model's ability to comprehend and reason about date-related information, such as calculating durations and interpreting dates.
- **CSQA** (CommonsenseQA) [3] dataset is designed to test a model's understanding of commonsense knowledge through multiple-choice questions. Each question requires reasoning over general world knowledge, with answer choices based on various plausible but nuanced options, testing the model's ability to pick the most commonsensical answer.
- **GSM8k** (Grade School Math 8K) [42] dataset is a collection of 8,000 challenging grade-school-level math word problems. It is designed to test a model's ability to perform multi-step arithmetic reasoning and solve math problems requiring logical thinking and numerical computation.

A.8 Note on ReConcile

ReConcile [34] is another relevant multi-agent framework that utilizes different LLMs with similar capabilities to engage in discussions and reach consensus. However, we exclude ReConcile from our baselines because its use of 3 LLMs of similar capabilities and makes it an unfair comparison to our framework, which employs 3 smaller models as peers and 1 larger model as the area chair. Simulating ReConcile with our setup would require excluding one of the models, either from the peer group or the area chair, which would lead to an

unbalanced debate. In particular, if we use 2 smaller models and a large model, the debate would be dominated by the larger model, resulting in biased outcomes. For these reasons, we do not include ReConcile as a direct baseline.

A.9 More Related Works

In this section we discuss more recent works in the field of multi-agent reasoning. While our ReFeR framework proposes a specific two-level hierarchy inspired by academic peer review, the broader field of multi-agent reasoning has explored several alternative collaborative architectures. Many recent works focus on achieving consensus through unstructured or semi-structured debate. For example, ReConcile [34] implements a "round-table conference" where diverse LLM agents iteratively discuss and refine their answers to reach a consensus, contrasting with ReFeR's single-pass hierarchical synthesis. Similarly, CONSENSAGENT [161] focuses on mitigating "sycophancy" in multi-agent debate through dynamic prompt refinement to achieve more efficient and effective consensus, whereas ReFeR's structure bypasses the need for peer consensus by vesting final authority in the area chair. Other frameworks introduce adaptive mechanisms to balance performance and efficiency. DOWN (Debate Only When Necessary) [54] proposes a dynamic framework that triggers a multi-agent debate only when an initial agent's confidence score is low, offering a different approach to efficiency than ReFeR-Lite's fixed, lightweight structure. Another direction involves diverse reasoning paths and role-playing. CoMM [35] prompts agents to adopt different expert roles, encouraging varied problem-solving strategies, which differs from ReFeR's functionally defined roles of peer evaluator and synthesizer. Lastly, some frameworks integrate external knowledge, such as MADKE [212], which provides agents with a shared knowledge pool to enhance their debating capabilities—a clear distinction from ReFeR's reliance on the internal knowledge of its constituent models. These kind of reasoning frameworks can be used for specific use cases where you have a defined knowledge pool.

A.10 Finetuning

Utilizing Analysis from larger LLMs ("Area Chair"), we enhance smaller LLMs through instruction-tuning, using a dataset crafted from comprehensive evaluations. We use the analysis feedback generated within the ReFeR framework, transforming it into a useful resource for instructional tuning. This fine-tuning significantly improves smaller models performance, enabling them to reach or surpass their larger counterparts in evaluation tasks. We use Mistral-7B, since it can be easily deployable on a small GPU and finetune. We used the instruction-tuning dataset (final output of Area Chair) of ReFeR framework as the training data.

Table A.4: Improving smaller models via instruction-tuning. Finetuning Results for Mistral-7B model on TopicalChat Dataset

Model	Coherence		Engagingness		Groundedness		Naturalness		Avg	
	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ
Mistral-7B No Finetune	0.124	0.102	0.167	0.134	0.078	0.069	0.100	0.081	0.117	0.096
Mistral-7B Finetuned	0.457	0.348	0.626	0.486	0.487	0.437	0.493	0.377	0.516	0.412

A.11 Communication Strategies:

Table A.5: Communication Strategies. Results on TopicalChat showing different generation and communication strategies for ReFeR-Pro.

Communication	Coherence		Engagingness		Groundedness		Naturalness		Avg	
Peer Feedback to AC	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ
Comment Only	**0.602**	**0.471**	0.635	0.502	**0.661**	**0.590**	0.587	0.454	0.621	0.504
Score Only	0.585	0.454	**0.673**	**0.535**	0.628	0.577	**0.625**	**0.484**	**0.628**	**0.513**
Both Comment & Score	0.580	0.453	0.642	0.512	0.605	0.545	0.555	0.427	0.596	0.484

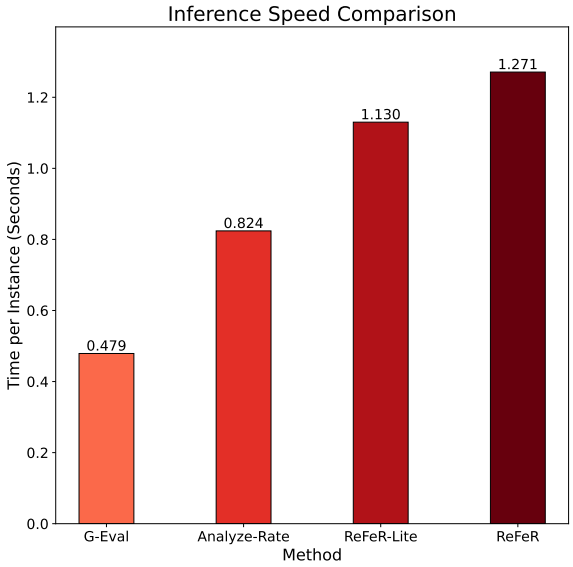
The type of feedback provided by peers to the area chair plays a crucial role in determining overall effectiveness. We explored three communication strategies: passing only scores, passing only comments, and passing both comments and scores. Table A.5 presents the impact of different feedback strategies on the framework's performance. The results indicate that passing only scores to the AC yields the best performance, with passing only comments being a close second. This is likely because when both comments and scores are passed, the AC model becomes more prone to confusion due to conflicting analyses or scores, and the longer prompt inputs negatively affect its decision-making [131]. Based on these findings, we adopt the scores-only strategy for all subsequent experiments with our framework.

A.12 Inference and Computation

Fig. A.4 presents the time taken per instance for ReFeR Variants and baseline models. G-Eval demonstrates the fastest inference speed, as it only generates scores. In contrast, Analyze-Rate takes nearly twice as long as G-Eval, since it produces both an analysis and a rating. ReFeR-Lite and ReFeR-Pro require only approximately 1.5 times the duration of Analyze-Rate, despite being multi-model frameworks. Notably, there is minimal difference between the Lite and Pro variants due to the influence of the n hyperparameter, indicating that the bulk of the processing time arises from the involvement of multiple models in the framework.

Although we show the monetary cost comparison of our framework vs the other methods, it is not the only cost for evaluation. We need to also consider test-time compute metric like

Figure A.4: Inference speed comparison with baselines



FLOPs processed. So as to know all parameter counts and calculate the FLOPs processed for each method, we use an open source model Qwen-2.5-72B, as our AC model (with the same peer models) and for G-Eval & Analyze-rate. We calculate the FLOPs with this approximation formula:

$$\text{FLOPs} = \text{layers} \times \left[4 \cdot \text{sequence\_length} \cdot d_{\text{model}}^2 + 8 \cdot d_{\text{model}} \cdot \text{ffn\_dim}\right]$$

The components involved in the FLOPs calculation are as follows: **layers** refers to the number of transformer layers, **sequence_length** is the length of the input sequence, **d_model** is the dimensionality of the model (hidden size), and **ffn_dim** represents the dimensionality of the feed-forward network, which is typically 4 times d_{model}.

For the self-attention mechanism, the cost of generating the query, key, and value matrices is given by:

$$3 \times d_{\text{model}}^2 \times \text{sequence\_length}.$$

The cost of the scaled dot-product attention, which involves computing attention scores and applying them, is approximately:

$$d_{\text{model}}^2 \times \text{sequence\_length}.$$

Therefore, the total cost for the self-attention mechanism is:

$$4 \times d_{\text{model}}^2 \times \text{sequence\_length}.$$

In the feed-forward network (FFN), which consists of two linear layers with a ReLU activation in between, the cost for each linear layer is approximately:

$$2 \times d_{\text{model}} \times \text{ffn\_dim}.$$

Thus, the total cost for the FFN, considering both the forward and backward passes, is:

$$8 \times d_{\text{model}} \times \text{ffn\_dim}.$$

Table A.6: Model Comparison of FLOPs. ReFeR-Lite is calculated as the cost of the peer models(first 3 rows) and AC. ReFeR-Pro is calculated as summation of the peers and 20 AC calls. G-Eval, Analyze-Rate are calculated as 20 calls.

Model	d_{model}	**ffn_dim**	**layers**	**Total Input tokens**	**FLOPs** ($\times 10^{15}$)
Llama-3.1-8B	4096	14336	32	970720	2.11
Mistral-Nemo-12B	5120	14336	40	970720	4.11
Gemma-2-9B	3584	28672	42	970720	2.14
AC (Qwen-2.5-72B) (n=1)	8192	29568	80	1016279	22.05
Analyze-Rate(n=20)	8192	29568	80	856100	372.16
G-Eval(n=20)	8192	29568	80	888500	386.07
ReFeR-Pro	8192	29568	80	-	449.31
ReFeR-Lite	8192	29568	80	-	30.40

Although the exact model architectures vary for different models, and hence the actual FLOPs vary, but we believe this approximation formula helps to give enough information for comparison of the computation. The given Total Input tokens is for all 4 dimensions(coherence,engagingness,etc) for each model/method. As mentioned, we use the same model Qwen-2.5 as our AC and for G-Eval, Analyze-Rate.

We modulate the performance with the hyperparameter *'n'* to see whether our method consistently is above the baselines in the performance-compute trade off and we present these results in Table A.7. But since the open models don't have 'n' hyperparamater, we do n calls to get the n responses.

For better understanding and visualization, we show the performance (spearman correlation) vs FLOPs in Figure A.5.

From Figure A.5 and Table A.7, we can observe that our ReFeR method is above the other methods in terms of performance-compute trade off. We can see that although ReFeR is relatively computationally costlier than the other methods, it has achieved higher correlations with relatively lower values of n, as we see that n=8 of ReFeR gives better performance than n=10 of G-eval while maintaining lesser FLOPs compared to G-Eval's n=10. We observe overall that ReFeR-Pro (or n=10 here,) has more FLOPS than G-Eval,

Table A.7: Performance and Computation of methods across different n-values. ρ is average spearman correlation across all metrics.

n	Method	ρ	FLOPs ($\times 10^{15}$)
1	ReFeR-Lite	0.620	30.41
	Analyze Rate	0.545	18.61
	G-Eval	0.608	19.3
3	ReFeR	0.639	74.51
	Analyze Rate	0.547	55.83
	G-Eval	0.626	57.9
5	ReFeR	0.646	118.61
	Analyze Rate	0.554	93.05
	G-Eval	0.633	96.5
8	ReFeR	0.648	184.76
	Analyze Rate	0.542	148.88
	G-Eval	0.636	154.4
10	ReFeR	0.649	230.5
	Analyze Rate	0.541	186.1
	G-Eval	0.637	193.0

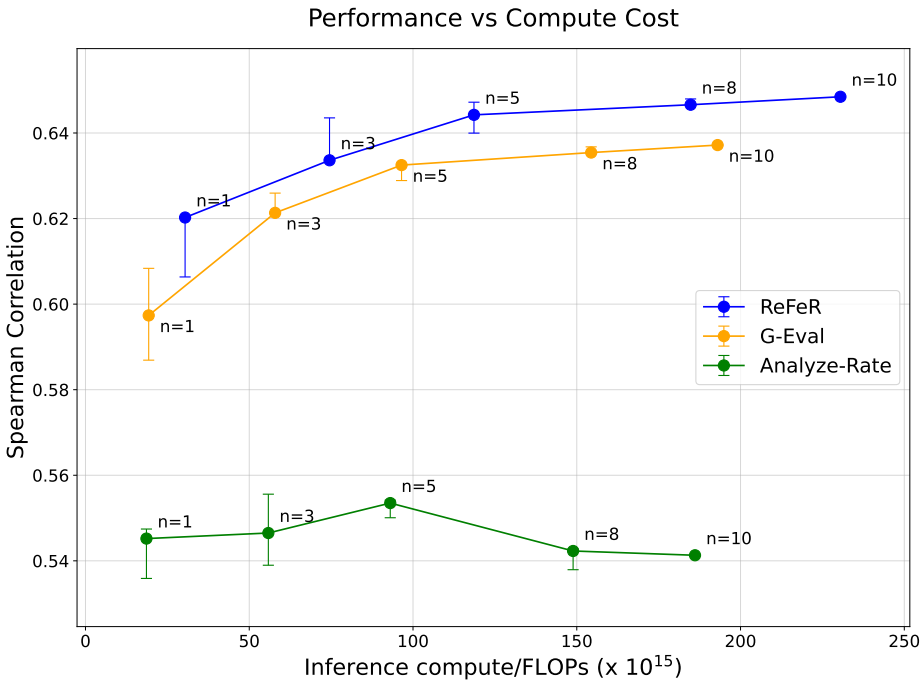


Figure A.5: Performance vs Compute Cost

but ReFeR gives more correlation than G-Eval. Whereas ReFeR-Lite (n=1) has significantly lesser FLOPs but gave very high correlation although didn't cross in this experiment with Qwen-2.5-72B model. But, we can also see some other observations here in this setting

with an open source model. Both ReFeR and G-Eval follow the same trend where we see an increase in performance with increase in *'n'*, but we don't see this trend in Analyze-rate. This leads us to the observation that some prompting schema are not consistent across different models.

Overall we do think that ReFeR-Lite is significantly better than other methods considering the overall FLOPs to performance ratio.

A.13 Statistical Tests

Table A.8: Statistical Tests. p-values for statistical tests between ReFeR Pro and baselines on TopicalChat.

Metric	ReFeR vs. Analyze rate	ReFeR vs. G-Eval
Coherence	2.34×10^{-6}	4.29×10^{-13}
Engagingness	3.70×10^{-5}	1.01×10^{-17}
Groundedness	1.15×10^{-7}	6.31×10^{-12}
Naturalness	1.19×10^{-4}	0.0736

We conducted the paired t-test to compare our method with the baseline and we report the p-values. As we can see for only one metric in G-Eval, the p-value is greater than the typical significance level (0.05). Hence, from the p-values, we can see that ReFeR is, in general, statistically significantly better than G-Eval and Analyze Rate.

A.14 Example of Reasoning improved by Framework

Example from GSM8K

Question: *Solo has to read 4 pages from his Science textbook, 20 pages from his Social Studies textbook, 7 pages from his History textbook and 8 pages from his Geography textbook. Solo read 15 pages on Monday. If he has 4 more days to complete all his reading, how many pages does he need to read, on average, in one day?*

Gold Answer: *6 pages*

Peer 1's Response: *To find the total number of pages Solo needs to read, we add up the pages from each textbook: 4 (Science) + 20 (Social Studies) + 7 (History) + 8 (Geography) = 39 pages. Solo needs to read, on average, 9.75 pages per day.* ***Answer: 9.75***

Peer 2's Response: *Total pages to read: 4 (Science) + 20 (Social Studies)*

+ 7 (History) + 8 (Geography) = 40 pages. After subtracting 15 pages already read, Solo needs to read 25 more pages. He needs to read an average of 6.25 pages per day. ***Answer: 6.25***

Peer 3's Response: *Total pages to read: 4 (Science) + 20 (Social Studies) + 7 (History) + 8 (Geography) = 39 pages. Solo needs to read, on average, 9.75 pages per day.* ***Answer: 9.75***

Area Chair's Response: *Total pages to read: 4 + 20 + 7 + 8 = 39 pages. After subtracting 15 pages already read, Solo needs to read 24 more pages. He needs to read an average of* ***6 pages*** *per day. LLM 1 and LLM 3 provided the correct calculations but made a mistake in their final answers by not subtracting the number of pages Solo read on Monday from the total number of pages. They both stated that Solo needs to read, on average, 9.75 pages per day, which is incorrect. LLM 2 made a calculation mistake. My solution corrects these errors and provides the accurate calculation for the average number of pages Solo needs to read per day.* ***Answer: 6***

Example 2 from GSM8K

Question: *Conor lives near a beach and loves going there every day to have fun. On a particular week, he found 50 people at the beach on the first day. The next day, 20 more people were present at the beach than on the first day, and on the third day, the total number of people increased to twice the number that were there on the second day. If Conor saw an average of 60 people each day the rest of the week, calculate the total number of people Conor saw at the beach that week.*

Gold Answer: *500*

Peer 1's Response:

Analysis: *To solve this problem, I will first determine the number of people present at the beach on each of the first three days. Then, I will calculate the total number of people Conor saw at the beach for the entire week by adding the number of people on each of the first three days to the average number of people he saw each day for the rest of the week. Since there are 5 more days in a standard week, I will multiply the average daily number of people by 5 to find the total number of people Conor saw for*

the remaining days. Finally, I will add the total number of people seen on the first three days to the total number of people seen during the rest of the week to find the overall total.
The first three days:
- Day 1: 50 people
- Day 2: 20 more people than on the first day, so 50 + 20 = 70 people
- Day 3: Twice the number of people on the second day, so 2 x 70 = 140 people
Total number of people on the first three days: 50 + 70 + 140 = 260
Average number of people on the rest of the week (5 days): 60 people/day
Answer: 560

Peer 2's Response:

Analysis: *On the first day, there were 50 people at the beach. On the second day, there were 20 more people than the first day, making it 70 people. On the third day, the number of people doubled from the second day, resulting in 140 people. For the remaining 4 days, Conor saw an average of 60 people each day, so the total for these days is 4 * 60 = 240 people. Adding all these together gives the total number of people Conor saw that week.* ***Answer: 450***

Peer 3's Response:

Analysis: *First, we calculate the number of people at the beach on the second and third days. The second day had 50 + 20 = 70 people. The third day had 70 * 2 = 140 people. Then we calculate the total number of people on the remaining five days as 60 * 5 = 300. Finally, we add up the people from all seven days: 50 + 70 + 140 + 300 = 560.*
Answer: 560

Area Chair's Response:
Analysis: *Let's break down the problem step by step.*

1. *On the first day, Conor saw 50 people at the beach.*
2. *On the second day, there were 20 more people than the first day, so the total for the second day is: 50 + 20 = 70 people.*

3. *On the third day, the number of people doubled compared to the second day, giving us:* $70 \times 2 = 140$ *people.*

4. *For the remaining four days of the week, Conor saw an average of 60 people each day. The total for these four days is:* $60 \times 4 = 240$ *people.*

Now, let's add all these numbers together to find the total number of people Conor saw at the beach over the week: 50 (day 1) + 70 (day 2) + 140 (day 3) + 240 (remaining days) = 500.
Ample insights were provided by the LLMs, particularly the correct calculation for each day. However, LLM 2 had a calculation mistake when reporting the final total as 450,which is inaccurate. LLM 3 correctly adds up the totals but does not account for the final calculation correctly, as it appears to calculate five days instead of the four remaining after the first three. Thus, compiling our information correctly gives us the right total: ***Answer: 500***

A.15 Hallucination

To understand the difference in response quality for our framework and the baselines, we used the HHEM-2.1 model by [18]. The HHEM-2.1 model measures the hallucination in the hypothesis (generated text) based on the premise (reference text). We did this analysis on the GSM8k experimental results, as the dataset provides the human annotated reasoning/rationale along with the answers. Hence we use this reference human reasoning as the premise and the generated reasoning in the final responses by different methods as the hypotheses. HHEM-2.1 gives scores in the range *(0,1)* where, 0 means that the hypothesis is not evidenced at all by the premise and 1 means the hypothesis is fully supported by the premise. Hence, the higher the HHEM score is, the lower the method's responses are hallucinated. We weren't able to test this on other benchmarks where human annotation reasoning/evaluation is not given.

We can see that ReFeR clearly outperforms all the baselines and GPT-4o-mini's(with our peer setup) hhem score. Considering the GPT-4o-mini's score as baseline, we can see that ReFeR has reduced hallucination, whereas the other baseline methods have increased hallucination significantly. This shows that ReFeR produces better rationale in responses compared to the baselines.

Table A.9: Comparison of average hallucination scores across methods on the GSM8k Benchmark. GPT-4o-mini is the HHEM score of GPT-4o-mini using our peer setup.

Method	HHEM Score
GPT-4o-mini	0.297
Zero-Shot-CoT	0.115
Self Correction	0.136
Multi-Agent Debate	0.102
Multi-Agent Peer Review	0.108
ReFeR	**0.330**

A.16 Homogeneous Experiments

We have demonstrated ReFeR primarily as a heterogeneous framework, where we leverage different models for peers and AC. Hence, we did this homogeneous experiment to test the framework's effectiveness when we use the same model for both the peers and AC. We use our best peer model Gemma2-9B as all peers & AC for this experiment, and to have diversity in responses among the peers and AC, we use the temperature hyperparameter, choosing temperatures *[0.25,0.5,0.75]* for the peer models. We retain the *temperature=1* for AC from our original setup. We can see the results for the homogeneous experiment in Table-A.10.

Table A.10: Performance of ReFeR framework with same model, Gemma2-9B as both peers and AC. Varying temperatures were used for diversity among peers.

Method	Coherence		Engagingness		Groundedness		Naturalness		Average	
	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ
Gemma (*temp=0.25*)	0.559	0.475	0.614	0.525	0.565	0.529	0.536	0.446	0.568	0.494
Gemma (*temp=0.5*)	0.548	0.465	0.611	0.525	0.571	0.535	0.540	0.452	0.568	0.494
Gemma (*temp=0.75*)	0.547	0.463	0.626	0.536	0.582	0.545	0.509	0.426	0.566	0.492
ReFeR-Pro	**0.587**	0.468	**0.681**	0.543	**0.628**	**0.572**	**0.597**	0.466	**0.623**	0.512
ReFeR-Lite	0.563	**0.476**	0.648	**0.557**	0.614	0.557	0.574	**0.480**	0.600	**0.517**

As we can see the framework has still shown to improve the performance using the same model for both peers and AC. Hence, we can see that the framework is effective even with using same model(varying temperatures) for peers and AC. This setting is particularly helpful when one wants to utilize a single model on the GPU for the entire evaluation.

A.17 Prompts

A.17.1 NLG Evaluation

TopicalChat

Coherence Peer Prompt

You will receive a dialogue between two people. Following that, there will be one suggested reply for the next part of the conversation, along with a related interesting fact.

Your job is to assess how coherent the suggested reply is, focusing on its ability to seamlessly continue the dialogue while also considering the overall context of the conversation, including the provided fact.

Please read and understand these instructions carefully. You may refer back to them as needed.

Assessment Criteria:

Coherence (1-3): How well does the response continue the conversation?
- A score of 1 (no) indicates that the reply significantly shifts the topic or disregards the ongoing conversation entirely.
- A score of 2 (somewhat) suggests that the response makes a vague reference to the conversation but fails to effectively engage with the dialogue or the accompanying fact.
- A score of 3 (yes) signifies that the response stays on topic, acknowledges the previous dialogue, and draws a clear and relevant connection to the interesting fact provided while maintaining the overall conversational flow.

Assessment Process:

1. *Review the conversation history for context and flow, focusing on how well the suggested reply relates to the previous exchanges.*
2. *Examine the suggested reply for its relevance and engagement with the ongoing dialogue.*
3. *Consider how well the reply connects with the interesting fact while also evaluating its contribution to the conversation as a whole.*
4. *Assign a coherence score of 1, 2, or 3, taking into account both the conversational progression and the connection to the fact.*

Example:

Conversation History: {{Conversation}}

Corresponding Fact: {{Contextual Fact}}

Response: {{Generated Response}}

Evaluation Form (Answer by starting with "Analysis:" to analyze the given example regarding the evaluation criteria as concise as possible, and then give the numeric rating on the next line by "Rating".)

Coherence:

Coherence AreaChair Prompt

Navigate through a simulated conversation between two individuals, followed by a provided potential response incorporating an intriguing fact. Your role is to assess the responses based on the coherence metric.

Alongside your evaluation, you will also receive initial evaluations from three large language models, referred to as the assistants' evaluations. Please read the instructions and criteria below carefully and use them as a guide in your evaluation, critically assessing the conversation, and the assistants' inputs.

Ensure a meticulous understanding of the instructions. Keep this document accessible for reference during the evaluation.

Evaluation Criteria:

Coherence (1-3): Assess whether the response seamlessly continues the conversation history.
- A score of 1 (no) denotes a significant shift in topic or disregard for the conversation history.
- A score of 2 (somewhat) indicates a response with limited reference to the conversation history and a noticeable shift in topic.
- A score of 3 (yes) signifies an on-topic response that strongly acknowledges and builds upon the conversation history.

Evaluation Steps:

1. *Thoroughly read the conversation history.*

2. Examine the potential response.

3. Evaluate coherence based on the conversation history.

4. Assign a coherence score of 1, 2, or 3.

Example:

Conversation History: {{Conversation}}

Corresponding Fact: {{Contextual Fact}}

Response: {{Generated Response}}

First Assistant's Evaluation: {{Peer_response1}}

Second Assistant's Evaluation: {{Peer_response2}}

Third Assistant's Evaluation: {{Peer_response3}}

Evaluation Form (Answer by starting with "Analysis:" to analyze the given example regarding the evaluation criteria as concise as possible, and then give the numeric rating on the next line by "Rating".)

Coherence:

Engagingness Peer Prompt

You will be given a conversation between two individuals. You will then be given one potential response for the next turn in the conversation. The response concerns an interesting fact, which will be provided as well.

Your task is to rate the responses on one metric. Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Engagingness (1-3): Is the response dull/interesting?
- A score of 1 (dull) means that the response is generic and dull.
- A score of 2 (somewhat interesting) means the response is somewhat interesting and could engage you in the conversation (e.g., an opinion, thought).
- A score of 3 (interesting) means the response is very interesting or presents an interesting fact.

Evaluation Steps:

1. *Read the conversation, the corresponding fact and the response carefully.*
2. *Rate the response on a scale of 1-3 for engagingness, according to the criteria above.*

Example:

Conversation History: {{Conversation}}

Corresponding Fact: {{Contextual Fact}}

Response: {{Generated Response}}

Evaluation Form (Answer by starting with "Analysis:" to analyze the given example regarding the evaluation criteria as concise as possible, and then give the numeric rating on the next line by "Rating".)

Engagingness:

Engagingness AreaChair Prompt

Navigate through a simulated conversation between two individuals, followed by a provided potential response incorporating an intriguing fact. Your role is to assess the responses based on the engagingness metric.

Alongside your evaluation, you will also receive initial evaluations from three large language models, referred to as the assistants' evaluations. Please read the instructions and criteria below carefully and use them as a guide in your evaluation, critically assessing the conversation, and the assistants' inputs.

Ensure a meticulous understanding of the instructions. Keep this document accessible for reference during the evaluation.

Evaluation Criteria:

Engagingness (1-3): Is the response dull or interesting?
- A score of 1 (dull) means that the response is generic and uninteresting.
- A score of 2 (somewhat interesting) means the response is somewhat engaging and

could capture interest (e.g., an opinion or thought).
- A score of 3 (interesting) means the response is highly engaging or presents an intriguing fact.

Evaluation Steps:

1. *Read the conversation, the corresponding fact, and the response carefully.*
2. *Rate the response on a scale of 1-3 for engagingness, according to the criteria above.*

Example:

Conversation History: {{Conversation}}

Corresponding Fact: {{Contextual Fact}}

Response: {{Generated Response}}

First Assistant's Evaluation: {{Peer_response1}}

Second Assistant's Evaluation: {{Peer_response2}}

Third Assistant's Evaluation: {{Peer_response3}}

Evaluation Form (Answer by starting with "Analysis:" to analyze the given example regarding the evaluation criteria as concisely as possible, and then give the numeric rating on the next line by "Rating".)

Engagingness:

Groundedness Peer Prompt

You will be given a conversation between two individuals. You will then be given one potential response for the next turn in the conversation. The response concerns an interesting fact, which will be provided as well.

Your task is to rate the responses on one metric. Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Groundedness (0-1) given the fact that this response is conditioned on, determine whether this response uses that fact.
- A score of 0 (no) means the response does not mention or refer to the fact at all.
- A score of 1 (yes) means the response uses the fact well.

Evaluation Steps:

1. *Read the conversation between the two individuals.*
2. *Identify the fact that is provided for the potential response.*
3. *Read the potential response.*
4. *Determine if the potential response uses or mentions the fact.*
5. *Assign a score of 0 or 1 for groundedness based on whether the response uses the fact.*

Example:

Conversation History: {{Conversation}}

Corresponding Fact: {{Contextual Fact}}

Response: {{Generated Response}}

Evaluation Form (Answer by starting with "Analysis:" to analyze the given example regarding the evaluation criteria as concise as possible, and then give the numeric rating on the next line by "Rating".)

Groundedness:

Groundedness AreaChair Prompt

Navigate through a simulated conversation between two individuals, followed by a provided potential response incorporating an intriguing fact. Your role is to assess the responses based on the groundedness metric.

Alongside your evaluation, you will also receive initial evaluations from three large language models, referred to as the assistants' evaluations. Please read the instructions and criteria below carefully and use them as a guide in your evaluation, critically

assessing the conversation, and the assistants' inputs.

Ensure a meticulous understanding of the instructions. Keep this document accessible for reference during the evaluation.

Evaluation Criteria:

Groundedness (0-1): Given the fact that this response is conditioned on, determine whether this response uses that fact.
- A score of 0 (no) means the response does not mention or refer to the fact at all.
- A score of 1 (yes) means the response uses the fact well.

Evaluation Steps:

1. *Read the conversation between the two individuals.*
2. *Identify the fact that is provided for the potential response.*
3. *Read the potential response.*
4. *Determine if the potential response uses or mentions the fact.*
5. *Assign a score of 0 or 1 for groundedness based on whether the response uses the fact.*

Example:

Conversation History: {{Conversation}}

Corresponding Fact: {{Contextual Fact}}

Response: {{Generated Response}}

First Assistant's Evaluation: {{Peer_response1}}

Second Assistant's Evaluation: {{Peer_response2}}

Third Assistant's Evaluation: {{Peer_response3}}

Evaluation Form (Answer by starting with "Analysis:" to analyze the given example regarding the evaluation criteria as concisely as possible, and then give the numeric rating on the next line by "Rating".)

Groundedness:

Naturalness Peer Prompt

You will be given a conversation between two individuals. You will then be given one potential response for the next turn in the conversation. The response concerns an interesting fact, which will be provided as well.

Your task is to rate the responses on one metric. Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Naturalness (1-3) Is the response naturally written??
- A score of 1 (bad) means that the response is unnatural.
- A score of 2 (ok) means the response is strange, but not entirely unnatural.
- A score of 3 (good) means that the response is natural.

Evaluation Steps:

1. *Read the conversation between the two individuals.*
2. *Read the potential response for the next turn in the conversation.*
3. *Evaluate the response based on its naturalness, using the provided criteria.*
4. *Assign a rating score of 1, 2, or 3 based on the evaluation.*

Example:

Conversation History: {{Conversation}}

Corresponding Fact: {{Contextual Fact}}

Response: {{Generated Response}}

Evaluation Form (Answer by starting with "Analysis:" to analyze the given example regarding the evaluation criteria as concise as possible, and then give the numeric rating on the next line by "Rating".)

Naturalness:

Naturalness AreaChair Prompt

Navigate through a simulated conversation between two individuals, followed by a provided potential response incorporating an intriguing fact. Your role is to assess the responses based on the naturalness metric.

Alongside your evaluation, you will also receive initial evaluations from three large language models, referred to as the assistants' evaluations. Please read the instructions and criteria below carefully and use them as a guide in your evaluation, critically assessing the conversation, and the assistants' inputs.

Ensure a meticulous understanding of the instructions. Keep this document accessible for reference during the evaluation.

Evaluation Criteria:

Naturalness (1-3): Is the response naturally written?
- A score of 1 (bad) means that the response is unnatural.
- A score of 2 (ok) means the response is strange, but not entirely unnatural.
- A score of 3 (good) means that the response is natural.

Evaluation Steps:

1. *Read the conversation between the two individuals.*
2. *Read the potential response for the next turn in the conversation.*
3. *Evaluate the response based on its naturalness, using the provided criteria.*
4. *Assign a rating score of 1, 2, or 3 based on the evaluation.*

Example:

Conversation History: {{Conversation}}

Corresponding Fact: {{Contextual Fact}}

Response: {{Generated Response}}

First Assistant's Evaluation: {{Peer_response1}}

Second Assistant's Evaluation: {{Peer_response2}}

Third Assistant's Evaluation: {{Peer_response3}}

Evaluation Form (Answer by starting with "Analysis:" to analyze the given example regarding the evaluation criteria as concisely as possible, and then give the numeric rating on the next line by "Rating".)

Naturalness:

A.17.2 MultiModal Evaluation

ICQD

Caption Quality Peer Prompt

Your task is to carefully evaluate the alignment between an image and its corresponding caption based on the provided criteria. Pay close attention to the instructions to ensure an accurate and nuanced assessment.

Instructions:

1. *Examine the image closely, identifying its key visual elements, objects, actions, and overall context.*
2. *Scrutinize the caption, comparing it to the visual content of the image, and identifying any inaccuracies, omissions, or misleading information. Consider both the explicit details and the overall context of the image.*
3. *Rate the caption on a scale of 1-100 according to the Evaluation Criteria, where 1 indicates a very poor match and 100 indicates a perfect match.*

Evaluation Criteria:

Rating (0-100): Evaluate the extent to which the caption aligns with the visual content of the image. A high rating should be given if the caption accurately reflects the main elements, actions, and context of the image, even if it uses concise language or omits minor details. Deduct points for inaccuracies, misleading descriptions, or significant omissions that distort the intended message of the image.

- 90-100: The caption perfectly or almost perfectly captures the image's content.
- 70-89: The caption is mostly accurate, with only minor inaccuracies or omissions.
- 50-69: The caption has notable inaccuracies or omissions but still partially represents the image.
- 30-49: The caption poorly represents the image, with significant inaccuracies or misleading elements.
- 0-29: The caption is almost entirely inaccurate or irrelevant to the image.

Example:

Image:

[Image will be provided separately]

Caption: {{Caption}}

Evaluation Form (Answer by starting with "Analysis:" to analyze the provided example regarding the evaluation criteria as concisely as possible, and then give the numeric rating on the next line by "Rating".)

Caption_ Quality:

Caption Quality AreaChair Prompt

You will be given an image, its caption, and you will also receive initial evaluations from two large language models, referred to as the assistants' evaluations.

Your task is to rate the caption on one metric.

Please read the instructions and criteria below carefully and use them as a guide in your evaluation.

Evaluation Criteria:

Relevance (0-100) - Assess how well the caption aligns with the content of the image. The caption should accurately describe or complement the visual elements and context of the image. Consider if the caption captures the key aspects of the image, its mood, and its intent, and whether it adds value by enhancing the viewer's understanding or experience of the image.

Evaluation Guidelines:

1. *Examine the Image: Carefully observe the image to understand its main elements, context, and message.*
2. *Review the Caption: Analyze if the caption accurately and effectively describes or complements the image. Consider the appropriateness of the language, tone, and whether the caption adds meaningful context or insight.*
3. *Rate the Caption's Relevance on a Scale of 0 to 100:*

   - *90-100: The caption is highly relevant, fully capturing the essence of the image with precise and insightful description or commentary, adding significant value to the image.*
   - *80-89: The caption is mostly relevant, capturing most key elements of the image with minor omissions or slightly less impactful language, still adding clear value.*
   - *70-79: The caption is somewhat relevant, capturing some key aspects but missing others, or includes minor irrelevant details, with a noticeable but limited enhancement to the image.*
   - *50-69: The caption has limited relevance, covering only a few elements of the image or providing a description that is either too generic or somewhat off-target, adding minimal value.*
   - *30-49: The caption is marginally relevant, with significant omissions or inaccuracies, possibly detracting from the image by misrepresenting it or providing little to no useful context.*
   - *10-29: The caption is largely irrelevant, missing the key aspects of the image, with significant inaccuracies or misrepresentations, adding no value or even confusing the viewer.*
   - *0-9: The caption is completely irrelevant or nonsensical, with no connection to the image, possibly confusing or misleading the viewer.*

   Example:

   Image:

   [Image is attached below]

   Caption: {{Caption}}

   First Assistant's Evaluation: {{Peer_Response1}}
Second Assistant's Evaluation: {{Peer_Response2}}

Evaluation Form (Answer by starting with "Analysis:" to analyze the provided example regarding the evaluation criteria, incorporating the peer ratings, and then give

the numeric rating on the next line by "Rating".)

Caption_ Quality:

AGIQA

Image Quality Peer Prompt

You will be given an image generated based on an input prompt.

Your task is to rate the image on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Image_ Quality (0-5) - the overall visual coherence and alignment with the input prompt. This rating should reflect how well the image matches the prompt, considering the clarity, relevance, and composition of the image.

Evaluation Steps:

- *Review the "Input Prompt" carefully to understand the intended content, theme, and style.*
- *Examine the generated image and compare it to the "Input Prompt". Check if the image accurately represents the prompt, is visually clear, and if the composition aligns with the expected outcome.*
- *Assign a score for Image Quality on a scale of 0 to 5, where 0 is the lowest and 5 is the highest based on the Evaluation Criteria.*

Example:

Input Prompt: {{Input_ Prompt}}

Generated Image:

[Image is attached below]

Evaluation Form (Answer by starting with "Analysis:" to analyze the given example regarding the evaluation criteria as concise as possible, and then give the numeric rating on the next line by "Rating".)

- Image_ Quality:

Image Quality AreaChair Prompt

You will be given an image generated based on an input prompt, along with initial evaluations from two assistants, referred to as the assistants' evaluations.

Your task is to rate the image on one metric.

Please read the instructions and criteria below carefully and use them as a guide in your evaluation.

Evaluation Criteria:

Image_ Quality (0-5) - Assess the visual coherence and alignment of the image with the input prompt. The image should reflect the content, theme, and style described in the prompt, and be visually clear and well-composed.

Evaluation Guidelines:

- *Review the "Input Prompt" to understand the intended content, theme, and style.*
- *Examine the generated image and analyze how well it represents the "Input Prompt" in terms of accuracy, clarity, and composition.*
- *Rate the image's quality on a scale of 0 to 5, with 0 being the lowest quality and 5 being the highest quality.*
- *Scoring Guidelines:*
  - *Score 5.0: The image fully captures the essence of the prompt with a high level of accuracy, clarity, and visual appeal, without any significant errors or irrelevant elements.*
  - *$4 \leq$ Score < 5: The image mostly aligns with the prompt, with minor inaccuracies or less relevant details, but still maintains a generally high quality.*
  - *$3 \leq$ Score < 4: The image partially represents the prompt, with noticeable inaccuracies or irrelevant details, and a less coherent visual presentation.*
  - *$2 \leq$ Score < 3: The image has significant deviations from the prompt, with major inaccuracies, irrelevant elements, and a disjointed visual composition.*
  - *$1 \leq$ Score < 2: The image fails to represent the prompt accurately, lacks visual coherence, and includes significant errors or irrelevant elements.*
  - *$0 \leq$ Score < 1: The image is completely unrelated to the prompt.*

Example:

Input Prompt: {{Input_ Prompt}}

Generated Image:

[Image is attached below]

First Assistant's Evaluation: {{Peer_ response1}}

Second Assistant's Evaluation: {{Peer_ response2}}

Please provide your analysis and rating as follows:

Evaluation Form (Answer by starting with "Analysis:" to analyze the given example regarding the evaluation criteria as concise as possible, and then give the numeric rating on the next line by "Rating".)

- Image_ Quality:

A.17.3 Reasoning

AQuA

AQuA Peer Prompt

You will be provided with a problem that requires logical reasoning, mathematical calculation, or both.

Your task is to solve the problem accurately, providing not just the correct answer but also a clear explanation of the steps taken to reach that answer.

It is crucial to thoroughly understand the problem and apply the correct principles or formulas to solve it.

Instructions:

- *Read the problem statement carefully, ensuring you understand all the details and what is required for the solution.*
- *Work through the problem logically and methodically, explaining your reasoning and the steps you take to solve the problem.*
- *Provide the final answer clearly, specifying it by choosing one of the provided options (e.g., A, B, C, etc.).*

Problem Statement: {{question}}

Provided Options: {{options}}

Evaluation Form:

- *Analysis: [Start with "Analysis:" to provide a concise and structured explanation of the steps and reasoning used to solve the problem. Ensure your analysis is clear and follows a logical sequence.]*
- *Answer: [Clearly state the final answer only (e.g., A, B, C, etc.) on the line after your analysis.]*

AQuA area chair Prompt

You will be provided with a problem that requires logical reasoning, mathematical calculation, or both. Along with the problem, you will also receive solutions from three other Language Models (LLMs).

Your task is to solve the problem accurately, using the peer responses to inform your approach. Apply the correct principles or formulas to arrive at the solution, while taking note of any useful insights or errors in the peer responses.

Instructions:

- *Understand the Problem: Read the problem statement carefully, ensuring you grasp all details.*
- *Review Peer Responses: Consider the solutions provided by the LLMs, noting useful approaches or any errors.*
- *Solve the Problem: Work through the problem logically, explaining your reasoning and steps. Utilize the peer responses as needed but ensure your solution is accurate and complete.*
- *Final Answer: Clearly state the final answer, choosing one of the provided options (e.g., A, B, C, etc.).*

Problem Statement: {{question}}

Provided Options: {{options}}

Solutions by Other LLMs:

- *LLM 1 Answer: {{Peer_response1}}*
- *LLM 2 Answer: {{Peer_response2}}*
- *LLM 3 Answer: {{Peer_response3}}*

Evaluation Form:

- *Analysis: [Start with "Analysis:", provide a concise explanation of your reasoning and steps, integrating relevant insights from the LLMs' responses.]*
- *Answer: [Clearly state the final answer label ONLY (e.g., A, B, C, etc.) on the line after your analysis. (DO NOT GIVE ANYTHING ELSE).]*

BBH_DU

BBH_DU Peer Prompt

You will be provided with a problem that requires understanding and interpreting dates or times logically.

Your task is to solve the problem accurately, providing not just the correct answer but also a clear explanation of the steps taken to reach that answer.

It is crucial to thoroughly understand the problem, applying the correct principles or formulas to arrive at the solution.

Instructions:

- *Read the problem statement carefully, ensuring you understand all the details and what is required for the solution.*
- *Work through the problem logically and methodically, explaining your reasoning and the steps you take to solve the problem.*
- *Provide the final answer clearly, specifying it by choosing one of the provided options (e.g., A, B, C, etc.).*

Problem Statement: {{question}}

Evaluation Form:

- *Analysis: [Start with "Analysis:" to provide a concise and structured explanation of the steps and reasoning used to solve the problem. Ensure your analysis is clear and follows a logical sequence.]*
- *Answer: [Clearly state the final answer only (e.g., A, B, C, etc.) on the line after your analysis.]*

BBH_DU AreaChair Prompt

You will be provided with a problem that requires understanding and interpreting dates or times logically. You will also receive the final answers from three other Language Models (LLMs).

Your task is to solve the problem accurately, using the answers provided by the LLMs to inform your reasoning. Provide a clear explanation of your approach, and arrive at your own final answer.

Instructions:

- *Understand the Problem: Read the problem statement carefully, ensuring you grasp all details.*
- *Review Peer answers: Consider the final answers provided by the LLMs, noting any patterns or outliers.*
- *Solve the Problem: Work through the problem logically, explaining your reasoning and steps. Use the peer answers as a reference but ensure your solution is accurate and complete.*
- *Final Answer: Clearly state the final answer, choosing one of the provided options (e.g., A, B, C, etc.).*

Problem Statement: {{question}}

Answers from Other LLMs:

- *LLM 1 answer: {{Peer_response1}}*
- *LLM 2 answer: {{Peer_response2}}*
- *LLM 3 answer: {{Peer_response3}}*

Evaluation Form:

- *Analysis: [Start with "Analysis:" to provide a concise explanation of your reasoning and steps to solve the problem, using the peer answers as a reference.]*
- *Answer: [Clearly state the final answer label ONLY (e.g., A, B, C, etc.) on the line after your analysis. (DO NOT GIVE ANYTHING ELSE).]*

CSQA

CSQA Peer Prompt

Evaluate the question by selecting the best option from the provided choices. Your task is to understand the context and nuances of the question, utilize your knowledge of the topic, and determine the most appropriate answer based on the options given. The goal is to select the most relevant and correct option that aligns with the question's intent.

Instructions:

- *Understand the Question: Read the question carefully to comprehend all aspects and the context in which it is asked.*
- *Consider the Options: Analyze each provided option carefully. Think about how each option relates to the question and the scenario it presents.*
- *Select the Best Option: Choose the option that best answers the question, based on your analysis. Focus on the logic or knowledge that supports this choice.*

Problem Statement: {{question}}

Provided Options: {{options}}

Evaluation Form:

- *Analysis: [Begin with "Analysis:" to provide a structured and clear explanation of your reasoning process. Your analysis should logically explain why the chosen option is the most appropriate answer to the question.]*
- *Answer: [Clearly state the final answer only (e.g., A, B, C, etc.) on the line after your analysis.]*

CSQA area chair Prompt

You will be provided with a question that requires careful evaluation to select the best option from the provided choices. You will also receive the final answers from three other Language Models (LLMs).

Your task is to determine the most appropriate answer, using the answers provided by the LLMs to inform your reasoning. Provide a clear explanation of your thought process and select the option that best aligns with the question's intent.

Instructions:

- *Understand the Question: Read the question carefully to comprehend all aspects and context.*
- *Review Peer Answers: Consider the final answers provided by the LLMs, noting any patterns or outliers.*
- *Select the Best Option: Based on your understanding and the peer answers, choose the option that best answers the question.*
- *Final Answer: Clearly state the final answer, choosing one of the provided options (e.g., A, B, C, etc.).*

Problem Statement: {{question}}

Provided Options: {{options}}

Answers from Other LLMs:

- *LLM 1 Answer: {{Peer_response1}}*
- *LLM 2 Answer: {{Peer_response2}}*
- *LLM 3 Answer: {{Peer_response3}}*

Evaluation Form:

- *Analysis: [Start with "Analysis:" to provide a concise and clear explanation of your reasoning, using the peer answers as a reference.]*
- *Answer: [Clearly state the final answer label ONLY (e.g., A, B, C, etc.) on the line after your analysis. (DO NOT GIVE ANYTHING ELSE).]*

GSM8k

GSM8k Peer Prompt

You will be provided with a problem that requires logical reasoning, mathematical calculation, or both.

Your task is to solve the problem accurately, providing not just the answer but also a clear explanation of the steps taken to reach that answer.

It is crucial to understand the problem thoroughly and apply the correct principles or formulas to solve it.

Instructions:

- *Read the problem statement carefully, ensuring you understand all the details and what is required for the solution.*
- *Work through the problem logically and methodically, explaining your reasoning and the steps you take to solve the problem.*
- *Provide the final answer clearly, specifying it as a numerical value or a specific explanation as required by the problem statement.*

Problem Statement: {{question}}

Evaluation Form:

- *Analysis: [Start with "Analysis:" to provide a concise and structured explanation of the steps and reasoning used to solve the problem. Ensure your analysis is clear and follows a logical sequence.]*
- *Answer: [Clearly state the final answer only (number) on the line after your analysis.]*

GSM8k area chair Prompt

You will be provided with a problem that requires logical reasoning, mathematical calculation, or both. You will also receive the final answers from three other Language Models (LLMs).

Your task is to solve the problem accurately, using the peer answers to inform your reasoning. Provide a clear explanation of your thought process and the steps taken to arrive at the solution. Ensure that your reasoning is sound and the final answer is

correct.

Instructions:

- *Understand the Problem: Read the problem statement carefully to ensure you grasp all the details and what is required.*
- *Review Peer Answers: Consider the final answers provided by the LLMs, noting any patterns or outliers.*
- *Work Through the Problem: Solve the problem methodically, using the peer answers as a reference. Explain your reasoning clearly.*
- *Final Answer: Provide the final answer clearly, specifying it as a numerical value or as required by the problem statement.*

Problem Statement: {{question}}

Answers from Other LLMs:

- *LLM 1 Answer: {{Peer_response1}}*
- *LLM 2 Answer: {{Peer_response2}}*
- *LLM 3 Answer: {{Peer_response3}}*

Evaluation Form:

- *Analysis: [Start with "Analysis:" to provide a concise and clear explanation of your reasoning, using the peer answers as a reference.]*
- *Answer: [Clearly state the final answer ONLY (number) on the line after your analysis. (DO NOT GIVE ANYTHING ELSE).]*