# Learning Inner Monologue and Its Utilization in Vision-Language Challenges

**Diji Yang**[1], **Kezhen Chen**[2], **Jinmeng Rao**[2], **Xiaoyuan Guo**[2], **Yawen Zhang**[2], **Jie Yang**[2], **Yi Zhang**[1]
[1]University of California, Santa Cruz, [2]Mineral
{dyang39,yiz}@ucsc.edu
{kezhenchen, jinmengrao, xiaoyuanguo, yawenz, yangjie}@mineral.ai

## Abstract

Inner monologue is an essential phenomenon for reasoning and insight mining in human cognition. In this work, we propose a novel approach for AI systems to simulate inner monologue. Specifically, we consider the communications between components in an LLM-centric system as inner monologues, and demonstrate inner monologue reasoning ability can be learned by supervised learning and reinforcement learning, and then be utilized to solve different complex vision-language problems in different domains. Driven by the power of Large Language Models (LLMs), two prominent methods for vision-language tasks have emerged: (1) the hybrid integration between LLMs and Vision-Language Models (VLMs), where visual inputs are firstly converted into language descriptions by VLMs, serving as inputs for LLMs to generate final answer(s); (2) visual feature alignment in language space, where visual inputs are encoded as embeddings and projected to LLMs' language space via further supervised fine-tuning. The first approach provides light training costs and interpretability but is hard to be optimized in an end-to-end fashion. The second approach presents decent performance, but feature alignment usually requires large amounts of training data and lacks interpretability. With inner monologue simulation, our approach achieves competitive performance with less training data and promising interpretability when compared with state-of-the-art models on two popular tasks.

## 1 Introduction

Recently, large language models (LLMs) have achieved substantial advancements. Notable models like PaLM (Chowdhery et al., 2022), InstructGPT (Ouyang et al., 2022), and LLaMA (Touvron et al., 2023) showcase their immense potential in the field of natural language processing and commonsense reasoning. Despite the impressive performance, LLMs are far from human-level reasoning in terms of multi-step reasoning ability and interpretability. When human perform complex reasoning, *inner monologue* plays an essential role in human cognition in which an individual engages in silent verbal communication with themselves. When people solve complicated reasoning problems, they tend to use "inner monologue" by performing reasoning via multi-turn self-conversations in their minds. Inner monologue helps people organize their thoughts and work through the optimal answers as a form of problem-solving (Cherney, 2023; Huang et al., 2022). To address the challenges in LLMs, we introduce a novel approach, **I**nner **M**onologue **M**ulti-Modal **O**ptimization (IMMO), to simulate the inner monologue process and specifically focus on complex reasoning in vision-language tasks, such as visual question-answering (VQA) or visual entailment (VE). Evidence shows that explicitly using natural language as the intermediate representation of reasoning is effective and essential for human cognition (Goldin-Meadow and Gentner, 2003; Chen et al., 2020; Lee et al., 2019; Zhang et al., 2023). Many researchers explore using natural language as the intermediate representation to bridge multiple modalities. Two research directions to add visual inputs into language space have been

NeurIPS Socially Responsible Language Modelling Research Workshop (NeurIPS-SoLaR 2023).

actively studied recently. The first direction is the hybrid integration between vision-language models (VLMs) and LLMs (Yang et al., 2022; Salaberria et al., 2023; Zhu et al., 2023a). Hybrid integration approaches aim to enable LLMs to utilize VLMs in a zero-shot or few-shot manner. These models do not require heavy training costs and provide interpretability as the model outputs from LLMs and VLMs are transparent. However, as LLMs do not access the visual inputs directly, they may miss some visual details in the images. Also, most hybrid integration approaches merge LLMs and VLMs in a discrete space, which are hard to optimize in an end-to-end fashion. The second direction is visual feature alignment in language space, where visual inputs are encoded as embeddings and projected to LLMs' language space via further supervised fine-tuning (Dai et al., 2023; Chen et al., 2023; Li et al., 2023a; Liu et al., 2023). This direction heavily relies on a large number of high-quality training data and lacks interpretability (Gao et al., 2022).

To tackle this dilemma, we enable an *Observer(s)* and a *Reasoner* to interact through natural language conversation by simulating inner monologue reasoning and propose to use supervised learning and reinforcement learning to learn how to perform the inner monologue. We choose one LLM as the Reasoner and one VLM as the Observer. Given a visual input and a visual reasoning problem, the Observer perceives the visual information and abstracts it into a natural-language description. The Reasoner decides whether the description has sufficient information to solve the problem or generates a further question for the Observer to acquire more visual information. With multiple turns, the Reasoner organizes the information and works through an answer. The figure in Appendix sec A shows two examples of solving visual questions with the inner monologue. To automatically learn the process of the inner monologue, the entire system is optimized by a policy gradient method named Proximal Policy Optimization (PPO) (Schulman et al., 2017). Explicitly modeling and providing the inner monologues make the thorough process for LM's decisions transparent, offer user insights into how it arrived at a particular output, help users understand why the model made certain choices, and prompt users or developers to identify and correct errors thereby improving models' reliability. As expected, it may also gain user trust, help users detect bias, support better human-AI collaboration, and help people study model behavior.

In summary, our contributions are as follows:

- Inspired by human cognition, we propose a novel approach, IMMO, to simulate inner monologue with LMMs and how we use it in vision-and-language reasoning. IMMO can be trained efficiently and has interpretability, which is also flexible to adapt to other modalities.
- We propose a two-stage training process to let the Observer(s) and Reasoner learn how to work together. We created a new human-like multi-turn inner monologue reasoning training corpus by augmenting existing VQA data with GPT-3.5.
- We evaluate IMMO on two vision-language reasoning tasks. Experiments show that IMMO can achieve competitive results compared with GPT-4 based hybrid integration approaches, while it uses significantly less training data and provides greater interpretability compared with embedding alignment approaches.

## 2 Related Works

The success of large pre-trained language models (LLMs) has led to significant advancements in solving vision-language problems by fusing visual representations into the language space. The essence of these works is to allow LLMs to understand information from other modalities, using their rich pre-trained language knowledge and the emerging ability (Wei et al., 2022a). Several recent works have explored two research directions: embedding alignment and hybrid integration. Approaches with embedding alignment focus on projecting visual embeddings to the language space and fusing vision and language information via supervised fine-tuning in the language space (Dai et al., 2023; Li et al., 2023a; Liu et al., 2023). Despite the impressive performance, these models demand extensive engineering efforts to collect the training data and lack interpretability. Our approach focuses on converting visual inputs to language descriptions while keeping decent performance, which provides more interpretability and reduces training costs significantly. Approaches involving hybrid integration convert visual inputs to language descriptions, such as image captions, via VLMs and solve problems with LLMs (Yang et al., 2022; Salaberria et al., 2023). However, this approach may lead to captions that are irrelevant to the question. To address this problem, several works adapt interactive multi-turn conversations to promote VLMs and LLMs interacting with each other and

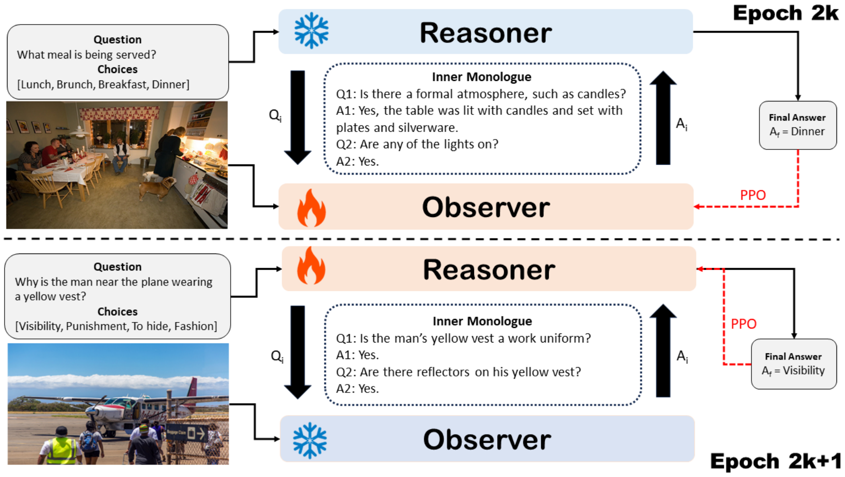

Epoch 2k
Reasoner
Question
What meal is being served?
Choices
[Lunch, Brunch, Breakfast, Dinner]
Q_i
Inner Monologue
Q1: Is there a formal atmosphere, such as candles?
A1: Yes, the table was lit with candles and set with plates and silverware.
Q2: Are any of the lights on?
A2: Yes.
A_i
Final Answer
A_f = Dinner
PPO
Observer
Reasoner
Question
Why is the man near the plane wearing a yellow vest?
Choices
[Visibility, Punishment, To hide, Fashion]
Q_i
Inner Monologue
Q1: Is the man's yellow vest a work uniform?
A1: Yes.
Q2: Are there reflectors on his yellow vest?
A2: Yes.
A_i
PPO
Final Answer
A_f = Visibility
Observer
Epoch 2k+1

Figure 1: The IMMO framework automatically acquires inner monologue capabilities through reinforcement learning. The Reasoner (LLM) and Observer (VLM) are alternately designated as the actively trainable model, highlighted with an orange hue and a fire icon, while the other model assumes the role of a static environmental representation, distinguished by a light blue shade and a snow icon.

acquiring more information  (You et al., 2023; Zhu et al., 2023a). Despite these works providing more interpretability and accessibility, they are usually in zero-shot or few-shot settings and have significantly lower performance compared to the embedding-alignment-based approaches. Our approach introduces a novel framework to optimize hybrid integration systems, which gives a more decent performance while preserving interpretability.

## 3 Approach

An overview of the IMMO framework for solving complex vision and language problems is shown in Figure 1. Our framework contains two components: Reasoner and Observer. The Observer takes images as the inputs and generates textual descriptions to describe the key information it observes. The Reasoner takes the generated textual descriptions and performs reasoning by either generating a new query for the Observer or generating the final results of the task. We choose an LLM as our Reasoner model and a VLM as our Observer model. The objective of the Reasoner is to generate effective queries to obtain targeted information and the objective of the Observer is to provide correct information based on the queries from the Reasoner. With multi-turn querying-answering conversations between the Reasoner and the Observer, the Reasoner gathers information to address the vision and language problems. Meanwhile, the Observer receives the queries and perceives more visual details from the image. In this section, we start by presenting the IMMO framework and then introduce the two-stage training approach for the IMMO framework.

### 3.1 Inner Monologue Multi-Modal Optimization

In the IMMO framework, the Reasoner and the Observer work together to solve a problem. Initially, the Observer receives the image $I$ and generates a caption $C$ to describe the basic visual information in the image. A text container $IM$ will be used here to track the inner monologue, including the initial caption $C$. At the intermediate $i$-th turn ($i \in [1, t]$ where t is the predefined maximum conversation turn), the Reasoner and the Observer will interact. Firstly, the Reasoner receives the original textual description of the problem/task $P$ combined with $IM_{i-1}$ and generates a query $Q_i$. Then, given $Q_i$, the Observer will provide the answer $A_i$ based on the image. At the end of each turn, both question and answer that are generated within the current turn will be added to the inner monologue history. After the final conversation turn $t$, the Reasoner will provide its prediction $A_f$ for the original problem $P$ based on all the collected inner monologue. During the multi-turn iteration, the Observer only accesses the input image and the most recent information query generated by the Reasoner, while the Reasoner can access the complete QA history at any given timestamp through the input prompt. Once the interaction reaches a pre-defined number of turns, the system prompts LLM for the final prediction. Appendix Sec. C describes the inner monologue process as formulas. Next, we describe how IMMO optimizes the Reasoner and the Observer jointly.

### 3.2 Two-Stage Training

Although this multi-turn conversational framework can be used in a zero-shot manner through prompting, the collaboration between the Reasoner and the Observer is suboptimal. The collaboration between the Reasoner and Observer should be further improved. For example, the Reasoner needs to be familiar with the Observer's capability in order to generate appropriate queries that the Observer can answer. The Observer, on the other hand, should be optimized to extract correct visual information based on the queries from the Reasoner. To alleviate the aforementioned underperformed collaboration problem, IMMO uses a two-stage training process: Supervised Human-prior Fine-tuning and Reinforcement Learning. Figure 1 shows the overview of the IMMO framework, and only the system-level reinforcement learning is illustrated.

To provide the Reasoner and the Observer a better starting point for reinforcement learning in the next stage, we employ supervised fine-tuning in the first stage, similar to the approach used in InstructGPT (Ouyang et al., 2022) to impart human reasoning patterns to the language model. Our training process focuses on imparting effective inner monologue to the model, going beyond simple chit-chat or prompt-based zero-shot learning. To achieve this, we enhance our pre-trained language model by introducing human prior knowledge and reasoning patterns with supervised fine-tuning. We utilize high-quality multi-turn conversational question-answering pairs annotated by humans as our training data. Both the Reasoner and the Observer are trained on the human-annotated data as a warm-up process. In the second stage, IMMO uses a special alternative training process for system-level reinforcement learning to jointly optimize multiple models while taking into account the dynamic interactions between models. Since the system involves two models, we use the alternating training strategy to prevent issues that may arise from updating two models simultaneously, such as the imbalance of capabilities of the Reasoner and the Observer (Goodfellow et al., 2014). Specifically, at the $2k$-th epoch where $k$ is a non-negative integer, we set the Observer as the active model (policy network) and the Reasoner as the environment model to provide feedback; at the $2k+1$-th epoch, we switch the Reasoner to be active and change the Observer as the environment model. During training, we only update the active model. Following the common approaches used in previous works (Stiennon et al., 2020; Ziegler et al., 2019) for fine-tuning auto-regressive decoder-only generative model, we treat the active model as the policy network. The active model is updated by PPO (Schulman et al., 2017) and the environment model remains frozen. Notably, the active model and the environment model only affect which model will be updated, and the input/output of each model strictly follows the multi-turn framework as shown in Figure 1. The algorithm uses the exact matching loss $r(A_f, G)$ between the predicted answer $A_f$ and the ground-truth answer $G$ as the major reward factor. The final reward $R$ shown in equation 1 also includes a KL penalty (Jaques et al., 2017) weighted by $\beta$ to ensure that the updated model $\mathcal{M}$ does not deviate too far from the well-trained starting point $\mathcal{M}_0$.

$$R = r(A_f, G) + \beta KL(\mathcal{M}, \mathcal{M}_0) \tag{1}$$

The training goal is to optimize the policy that maximizes the expected reward. Appendix Sec. D shows the algorithm of the overall training procedure.

## 4 Experiment

To evaluate the effectiveness of IMMO for complex vision-language reasoning, we conducted experiments on two popular tasks: Commonsense Visual Question Answering (VQA) and Visual Entailment (VE). Both tasks require models to have commonsense knowledge and reasoning abilities. This section first describes our implementation of the IMMO framework and then presents the details of these two tasks. We also provide a detailed comparison between IMMO with existing VQA methods in Appendix Sec. F and an ablation study of the impact of conversation turns in Appendix Sec. G. Appendix Sec. H presents the details of data preparation and model implementation.

[1] All the experiments in the table are named after the methods, and the language models involved are all Vicuna-7B.

Table 1: Results on ScienceQA test set.

| Method [1] | Training | Reasoning Aids | Accuracy(%) |
|---|---|---|---|
| PICa | Zero-shot | None | 54.3 |
| Vicuna | 16-shots | Chain-of-Thought | 68.6 |
| IMMO | 16-shots | Inner-Monologue | 74.0 |
| Vicuna | SL | Chain-of-Thought | 78.3 |
| IMMO | SL+RL | Inner-Monologue | **84.8** |

**Commonsense Visual Question Answering** We conducted experiments on the ScienceQA (SQA) (Lu et al., 2022) dataset, which is a standard benchmark for commonsense VQA, collected from elementary and high school exams. Appendix Sec. E shows some success and failure examples of our method. To study the impact of RL optimization and inner monologue on model performance, we conduct experiments with 3 baselines. As shown in Table 1, different training methods and reasoning aids were examined, while we used Vicuna-7B as LLM and BLIP-2 as a captioning model in all cases. The first baseline is PICa (Yang et al., 2022). For a fair comparison, instead of using the default GPT-3 as LLM in PICa, we let PICa use Vicuna as its LLM. The second baseline is Vicuna-16-shots, which incorporates the Chain-of-Thoughts (CoT) prompting (Wei et al., 2022b) with the Vicuna model. The third baseline is Vicuna-SL, which is LLM fined tuned with the training data following instruction fine-tuning (Chung et al., 2022; Taori et al., 2023). We also compare two training methods for IMMO: few-shot learning (IMMO 16-shots) vs. two-stage training described in Section 3.2 (IMMO SL+RL). Following the baseline setting, table 1 presents our results on the SQA test set. Results show that our approach outperforms both few-shot baselines and the supervised finetuning baseline.

Table 2: Results on SNLI-VE.

| | Method | Accuracy(%) |
|---|---|---|
| Emb | MiniGPT4 (Zhu et al., 2023b) | 35.1 |
| | LLaVA (ZS) (Liu et al., 2023) | 40.3 |
| | OFA (Wang et al., 2022) | 91.0 |
| NL | IdealGPT (You et al., 2023) | 55.3 |
| | Vicuna-16-shots | 49.8 |
| | Vicuna-SL | 59.8 |
| | IMMO | 65.7 |

**Visual Entailment** SNLI-VE (Xie et al., 2018) is a widely-used task designed as a classification problem for vision-language reasoning: identify whether the relationship between the given image premise and text hypothesis is entailment, neural, or contradiction. Table 2 shows the results on the SNLI-VE dev set. We add another baseline: IdealGPT (You et al., 2023), a recent hybrid integration approach that utilizes the rich reasoning knowledge of GPT-3.5-175B. Among approaches that use text to represent visual information, IMMO achieves the best performance. With a much smaller LLM, our best-performing checkpoints trained from Vicuna-7B achieved a 10.4% improvement over IdealGPT (65.7% vs 55.3%). The results of 3 embedding-based methods (MiniGPT4 (Zhu et al., 2023b), LLaVA (Liu et al., 2023), and OFA (Wang et al., 2022)) are also reported as a reference. While well-tuned embedding-based methods such as OFA work well on this dataset, single-model-based end-to-end optimization is not practical when interpretability is required or the training cost is too high. In such situations, hybrid integration like IMMO could be a good choice.

## 5 Conclusion

Inspired by cognitive modeling, we apply inner monologue, a commonly seen human reasoning process, in the interaction between LLM and VLM. The Inner Monologue explicitly visualizes the system's problem-solving progress in a human-readable way, which improves the interpretability while maintaining decent performance.

## References

Chen, F.; Han, M.; Zhao, H.; Zhang, Q.; Shi, J.; Xu, S.; and Xu, B. 2023. X-LLM: Bootstrapping Advanced Large Language Models by Treating Multi-Modalities as Foreign Languages. *arXiv preprint arXiv:2305.04160*.

Chen, K.; Huang, Q.; Palangi, H.; Smolensky, P.; Forbus, K.; and Gao, J. 2020. Mapping Natural-language Problems to Formal-language Solutions Using Structured Neural Representations. *ICML*.

Cherney, K. 2023. Everything to Know About Your Internal Monologue. *Healthline*.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; et al. 2022. PaLM: Scaling Language Modeling with Pathways. *arXiv preprint arXiv:2204.02311*.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, E.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Dai, W.; Li, J.; Li, D.; Tiong, A. M. H.; Zhao, J.; Wang, W.; Li, B.; Fung, P.; and Hoi, S. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*.

Gao, F.; Ping, Q.; Thattai, G.; Reganti, A.; Wu, Y. N.; and Natarajan, P. 2022. Transform-Retrieve-Generate: Natural Language-Centric Outside-Knowledge Visual Question Answering. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5057–5067.

Goldin-Meadow, S.; and Gentner, D. 2003. *Language in mind: Advances in the study of language and thought*. MIT Press.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Huang, W.; Xia, F.; Xiao, T.; Chan, H.; Liang, J.; Florence, P.; et al. 2022. Inner Monologue: Embodied Reasoning through Planning with Language Models. *arXiv preprint arXiv:2207.05608*.

Jaques, N.; Gu, S.; Bahdanau, D.; Hernández-Lobato, J. M.; Turner, R. E.; and Eck, D. 2017. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *International Conference on Machine Learning*, 1645–1654. PMLR.

Khashabi, D.; Min, S.; Khot, T.; Sabharwal, A.; Tafjord, O.; Clark, P.; and Hajishirzi, H. 2020. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*.

Lee, K.; Palangi, H.; Chen, X.; Hu, H.; and Gao, J. 2019. Learning Visual Relation Priors for Image-Text Matching and Image Captioning with Neural Scene Graph Generators. *arXiv preprint arXiv:1909.09953*.

Li, B.; Zhang, Y.; Chen, L.; Wang, J.; Yang, J.; and Liu, Z. 2023a. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023b. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

Lu, P.; Mishra, S.; Xia, T.; Qiu, L.; Chang, K.-W.; Zhu, S.-C.; Tafjord, O.; Clark, P.; and Kalyan, A. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35: 2507–2521.

Lu, P.; Peng, B.; Cheng, H.; Galley, M.; Chang, K.-W.; Wu, Y. N.; Zhu, S.-C.; and Gao, J. 2023. Chameleon: Plug-and-play compositional reasoning with large language models. *arXiv preprint arXiv:2304.09842*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.

Salaberria, A.; Azkune, G.; de Lacalle, O. L.; Soroa, A.; and Agirre, E. 2023. Image captioning for effective use of language models in knowledge-based visual question answering. *Expert Systems with Applications*, 212: 118669.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Schwenk, D.; Khandelwal, A.; Clark, C.; Marino, K.; and Mottaghi, R. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*, 146–162. Springer.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021.

Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Stanford Alpaca: An Instruction-following LLaMA model. `https://github.com/tatsu-lab/stanford_alpaca`.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

von Werra, L.; Belkada, Y.; Tunstall, L.; Beeching, E.; Thrush, T.; and Lambert, N. 2020. TRL: Transformer Reinforcement Learning. `https://github.com/lvwerra/trl`.

Wang, P.; Yang, A.; Men, R.; Lin, J.; Bai, S.; Li, Z.; Ma, J.; Zhou, C.; Zhou, J.; and Yang, H. 2022. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, 23318–23340. PMLR.

Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Chi, E.; Le, Q.; and Zhou, D. 2022b. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Online: Association for Computational Linguistics.

Xie, N.; Lai, F.; Doran, D.; and Kadav, A. 2018. Visual entailment task for visually-grounded language learning. *arXiv preprint arXiv:1811.10582*.

Yang, Z.; Gan, Z.; Wang, J.; Hu, X.; Lu, Y.; Liu, Z.; and Wang, L. 2022. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36:3, 3081–3089.

You, H.; Sun, R.; Wang, Z.; Chen, L.; Wang, G.; Ayyubi, H.; Chang, K.-W.; and Chang, S.-F. 2023. IdealGPT: Iteratively Decomposing Vision and Language Reasoning via Large Language Models. *arXiv preprint arXiv:2305.14985*.

Zhang, Z.; Zhang, A.; Li, M.; Zhao, H.; Karypis, G.; and Smola, A. 2023. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*.

Zhu, D.; Chen, J.; Haydarov, K.; Shen, X.; Zhang, W.; and Elhoseiny, M. 2023a. ChatGPT Asks, BLIP-2 Answers: Automatic Questioning Towards Enriched Visual Descriptions. *arXiv preprint arXiv:2303.06594*.

Zhu, D.; Chen, J.; Shen, X.; Li, X.; and Elhoseiny, M. 2023b. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

Ziegler, D. M.; Stiennon, N.; Wu, J.; Brown, T. B.; Radford, A.; Amodei, D.; Christiano, P.; and Irving, G. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

## Social Impacts Statement

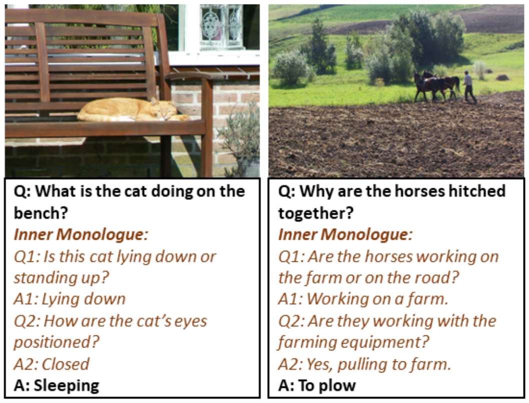
The most popular trend of multi-modal reasoning recently is projecting visual embeddings to the language space which is optimized via multi-modal instruct tuning. However, this direction requires heavy training costs (resources, electric energy, high-quality annotated data) and lacks interpretability. Our approach is inspired by human cognition and provides an alternative way for multi-modal reasoning via trainable inner monologue, which significantly reduces training costs while keeping decent performance and interpretability. With these features, IMMO is more practical for real-world cases and also saves more energy.

This paper is a first step towards this research direction, and there is much room for future improvement. Our current implementation promotes the reasoner querying certain turns, while an ideal reasoner should autonomously determine whether to continue querying or end the inner monologue with direct answers (for example, when adequate information has been gathered or due to time/resource constraints). Our implementation only includes one observer, while it's possible to include more observers with different modalities or functionalities. Due to the resource limits, we used a synthetic way to generate supervised training data, while organizations with ample resources could hire human annotators to provide more labeled data with higher quality. The reward function could also be further studied.

# Appendix

## A Example of Multi-modal Inner Monologue

Figure 2 shows two examples of solving visual questions with the inner monologue.



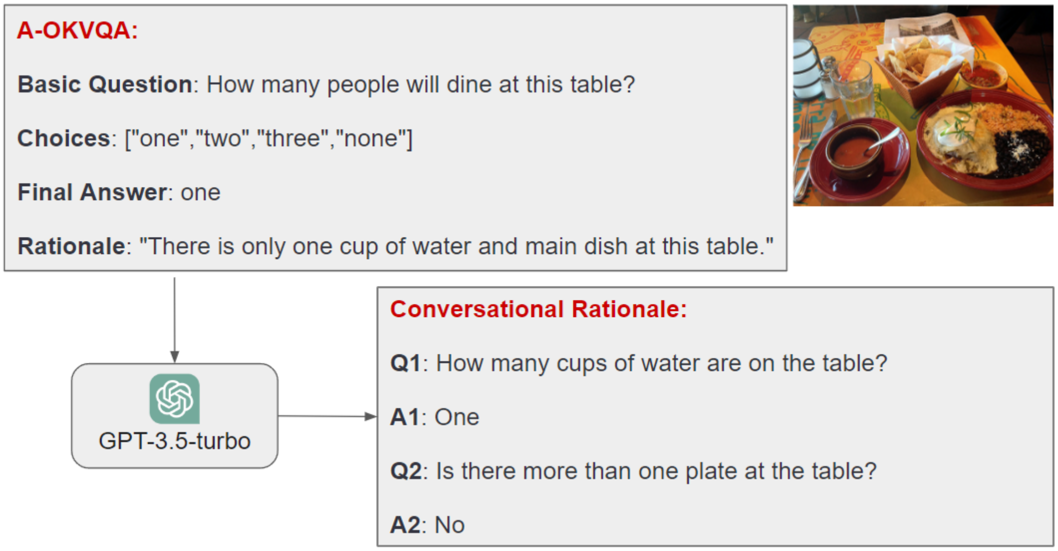
**Q: What is the cat doing on the bench?**
*Inner Monologue:*
*Q1: Is this cat lying down or standing up?*
*A1: Lying down*
*Q2: How are the cat's eyes positioned?*
*A2: Closed*
**A: Sleeping**

**Q: Why are the horses hitched together?**
*Inner Monologue:*
*Q1: Are the horses working on the farm or on the road?*
*A1: Working on a farm.*
*Q2: Are they working with the farming equipment?*
*A2: Yes, pulling to farm.*
**A: To plow**

Figure 2: Examples of multi-model inner monologue.

## B Example of Inner Monologue Warmup data

Figure 3 demonstrates the procedure for the data augmentation. Please see the data appendix for more data examples.



**A-OKVQA:**

**Basic Question**: How many people will dine at this table?

**Choices**: ["one","two","three","none"]

**Final Answer**: one

**Rationale**: "There is only one cup of water and main dish at this table."

**Conversational Rationale:**

**Q1**: How many cups of water are on the table?

**A1**: One

**Q2**: Is there more than one plate at the table?

**A2**: No

Figure 3: Example of converting human written declarative rationale to dialogue form reasoning path.

## C Inner Monologue Formulas

In the IMMO framework, the Reasoner and the Observer work together to solve a problem. Initially, the Observer receives the Image $I$ and generates a caption $C$ to describe the basic visual information in the image. A text container $IM$ will be used here to track the inner monologue, including the initial caption $C$:

$$IM_0 = C = Observer(I) \tag{2}$$

At the intermediate $i$-th turn ($i \in [1, t]$ where t is the predefined maximum conversation turn), the Reasoner and the Observer will interact. Firstly, the Reasoner receives the original textual description of the problem/task $P$ combined with $IM_{i-1}$ and generates a query $Q_i$. Then, given $Q_i$, the Observer will provide the answer $A_i$ based on the image. This process is shown as the equations 3, and 4.

$$Q_i = Reasoner(P, IM_{i-1}) \tag{3}$$

$$A_i = Observer(I, Q_i) \tag{4}$$

At the end of each turn, both question and answer that are generated within the current turn will be added to the inner monologue history. The $IM_i$ is defined as:

$$IM_i = C + \sum_{j=0}^{i}(Q_j + A_j) \tag{5}$$

After the final conversation turn $t$, the Reasoner will provide its prediction $A_f$ for the original problem $P$ based on all the collected inner monologue:

$$A_f = Reasonser(P + IM_t) \tag{6}$$

During the multi-turn iteration, the Observer only accesses the input image and the most recent information query generated by the Reasoner, while the Reasoner can access the complete QA history at any given timestamp through the input prompt. Once the interaction reaches a pre-defined number of turns, the system prompts LLM for the final prediction.

## D Algorithm of IMMO

The overall training procedure of IMMO is shown in Algorithm 1.

**Algorithm 1** IMMO Reinforcement Learning

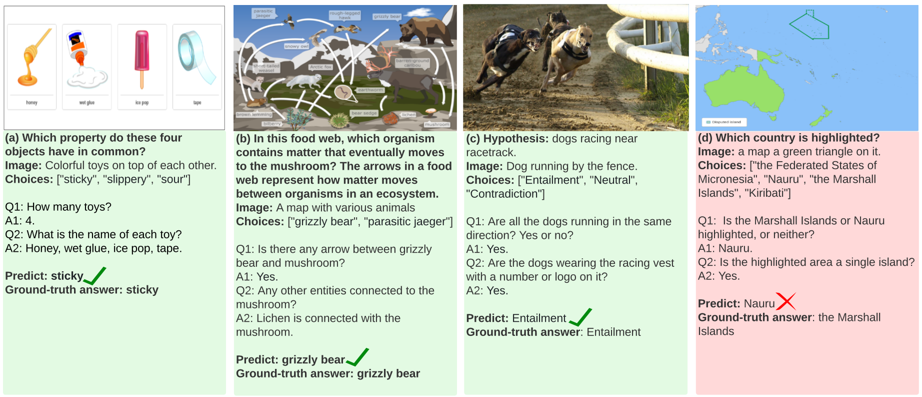
```
Dataset: (Problem $P$, Image $I$, Ground Truth $G$) tuples
Reasoner: a pre-trained large language model
Observer: a pre-trained vision-language model
N: training epoch
t: pre-defined max turns
k: any none-negative integer
for epoch = 1 to N do
    Set $Reasoner$ as the active model $\mathcal{M}$
    Set $Observer$ as the environment model $\mathcal{E}$
    if epoch = 2k then
        Set $Observer$ as the active model $\mathcal{M}$
        Set $Reasoner$ as the environment model $\mathcal{E}$
    end if
    Sample ($P$, $I$, $G$) from the dataset
    $C \leftarrow Observer(I)$
    Set $IM_0 = C$
    for $i = 1$ to $t$ do
        $Q_i \leftarrow Reasoner(P, IM_{i-1})$
        $A_i \leftarrow Observer(I, Q_i)$
        $IM_i = IM_{i-1} + Q_i + A_i$
    end for
    $A_f = Reasoner(P, IM_t)$
    Reward $\leftarrow \mathcal{R}$ {Eq. 1}
    Update $\mathcal{M}$ using PPO
end for
```

## E Representative cases from IMMO

Figure 4 displays instances of successful outcomes (a, b, and c) as well as an unsuccessful case (d) depicting the interpretable inner monologues generated by IMMO. Example (b) shows LLM's ability to compensate for VLM inaccuracies (incorrect A2) through reasoning and available information. Moreover, the questioning path in (b) and (c) demonstrate LLM's vigilance in monitoring VLM responses, persisting in using subsequent questions Q2 to validate information after Q1, even when the information is enough to answer the main question. On the other side, example (d) exposes how the VLM's limited geographical background knowledge hinders LLM from arriving at an accurate answer. However, the erroneous visual information from VLM misleads the LLM into an incorrect final prediction. These examples illustrate the interpretability of IMMO.

## F Comparison with additional VQA methods

We did some further analysis to compare representative solutions on the ScienceQA task (Table 3). The embedding alignment approach, LLaVA (Liu et al., 2023), performs well; however, it requires extensive training data and lacks interpretability. Hybrid integration such as IMMO, Chamaleon (Lu



**(a) Which property do these four objects have in common?**
**Image:** Colorful toys on top of each other.
**Choices:** ["sticky", "slippery", "sour"]

Q1: How many toys?
A1: 4.
Q2: What is the name of each toy?
A2: Honey, wet glue, ice pop, tape.

**Predict: sticky** ✓
**Ground-truth answer: sticky**

**(b) In this food web, which organism contains matter that eventually moves to the mushroom? The arrows in a food web represent how matter moves between organisms in an ecosystem.**
**Image:** A map with various animals
**Choices:** ["grizzly bear", "parasitic jaeger"]

Q1: Is there any arrow between grizzly bear and mushroom?
A1: Yes.
Q2: Any other entities connected to the mushroom?
A2: Lichen is connected with the mushroom.

**Predict: grizzly bear** ✓
**Ground-truth answer: grizzly bear**

**(c) Hypothesis:** dogs racing near racetrack.
**Image:** Dog running by the fence.
**Choices:** ["Entailment", "Neutral", "Contradiction"]

Q1: Are all the dogs running in the same direction? Yes or no?
A1: Yes.
Q2: Are the dogs wearing the racing vest with a number or logo on it?
A2: Yes.

**Predict:** Entailment ✓
**Ground-truth answer**: Entailment

**(d) Which country is highlighted?**
**Image:** a map a green triangle on it.
**Choices:** ["the Federated States of Micronesia", "Nauru", "the Marshall Islands", "Kiribati"]

Q1:  Is the Marshall Islands or Nauru highlighted, or neither?
A1: Nauru.
Q2: Is the highlighted area a single island?
A2: Yes.

**Predict:** Nauru ✗
**Ground-truth answer**: the Marshall Islands

Figure 4: Success and failure examples of IMMO.

| | LLaVa | Chameleon | UnifiedQA | IMMO |
|---|---|---|---|---|
| Interpretable | ✗ | ✓ | ✓ | ✓ |
| Trainable | ✓ | ✗ | ✓ | ✓ |
| Model size | 13B | GPT-4 | 223M | 9B |
| Tuned param | 13B | 0 | 223M | 5M |
| Data usage | 770K | - | 17K | 25K |
| SQA | 90.9 | 86.5 | 74.1 | 84.8 |

Table 3: Comparison with other ScienceQA approaches.

et al., 2023) and UnifiedQA (Khashabi et al., 2020; Lu et al., 2022) employ modular architecture, enabling model-wise interpretability by access to individual outputs from sub-modules. Chameleon is based on GPT-4, which is not publicly available, and poses constraints on its adoption and further fine-tuning. Compared with Chamaleon, IMMO achieved comparable performance with a significantly smaller model. UnifiedQA uses supervised training akin to our Vicuna-SL baseline, however, it falls short due to the lack of system-wide optimization and information loss when converting images to captions. Compared with UnifiedQA, IMMO addresses these problems via inner monologue and two-stage training, which significantly improves the performance of the hybrid integration method. Furthermore, compared to Chamaleon and UnifiedQA, which offers simple model-level interpretability, IMMO's entire complex multi-round reasoning procedure is also transparent and human-readable.

Notably, we can not rule out the possibility that black-box models like GPT-4 might have inadvertently or intentionally undergone training using publicly accessible test data. Thus we list the performance of those black-box models here for reference instead of as a baseline for a fair comparison. We expect the proposed approach could be applied to other LLMs and VLMs such as GPT-4 and further improve their performance.

## G The Impact of Conversation Turns

To examine the impact of inner monologue turns on performance, we conduct ablation tests on ScienceQA using both few-shot and trained approaches. Maintaining constant hyperparameters, we evaluate turns ranging from 0 to 5, where 0 means VLM only provides an initial caption. As shown in Figure 5, accuracy notably rises on the SQA test set from 0 to 2 turns, plateauing thereafter. Our analysis identifies SQA questions as demanding less multi-hop reasoning than background knowledge. Thus, LLM's primary learned strategy involves querying key facts initially, followed by confirmation or asking for side information. Also, more conversations bring uncertainty to the interaction as LLM may ask less relevant questions after 3 turns or VLM brings incorrect visual information. This trend is accentuated under the few-shot setting. Without training, LLM appears to be less robust to noise conversation, so the performance rapidly decreases after 2 turns. It's important to note that these

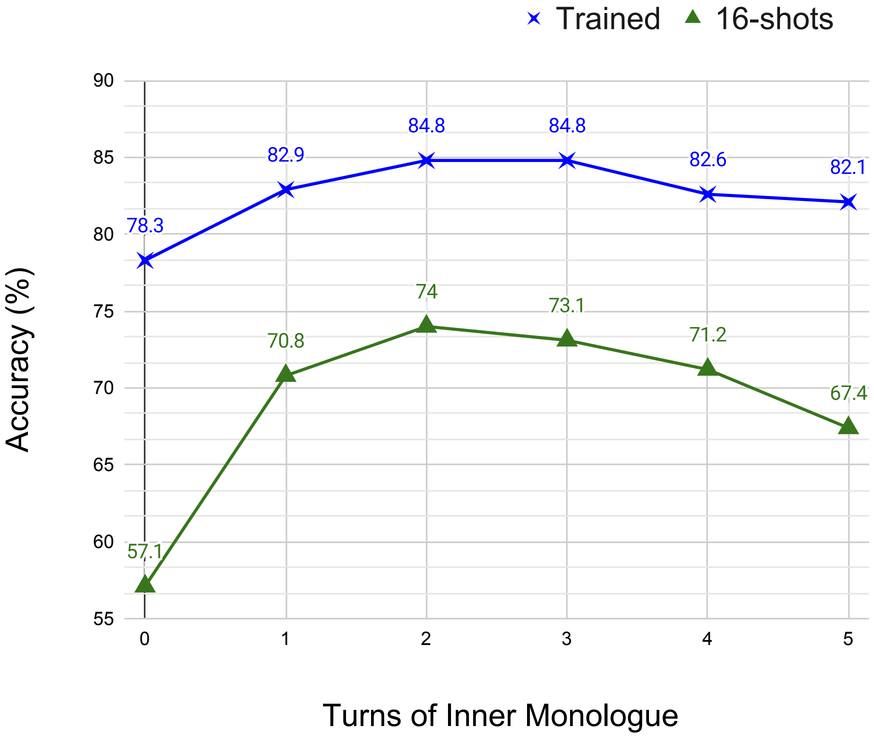


Figure 5: Ablation study on different inner monologue turns using on ScienceQA test set under few-shot and trained manner.

findings are specific to ScienceQA question patterns, underscoring the best inner monologue turns are highly based on the dataset's characteristics.

## H Data and Implementation

We construct a new training corpus for supervised human-prior fine-tuning by utilizing the A-OKVQA (Schwenk et al., 2022) dataset, which includes human-annotated reasoning paths labeled as rationale. We derive inner monologue from rationale. By prompting GPT-3.5 in a zero-shot manner, we transform rationale into two-turn question-answering pairs (examples are Appendix Sec. B). The results are then combined with 17k single-turn VQA samples. Each sample in the training corpus contains a question, a choice list, two rounds of QA conversations, and the correct answer. At the supervised fine-tuning stage, we optimize the autoregressive LLM by performing the next token prediction task over this augmented corpus. At the reinforcement learning stage, the training is mainly based on the Transformers-Reinforcement-Learning (TRL) solution (von Werra et al., 2020) to wrap up the Hugging Face trainer (Wolf et al., 2020). For different tasks (VQA or VE), reinforcement learning is performed on task-specific training sets.

Our proposed system uses the Vicuna-7b (Chiang et al., 2023) language model and BLIP-2 (Li et al., 2023b) vision-language model. To ensure computational efficiency, we employed the Low-rank adaptation (Lora) (Hu et al., 2021) to train only 0.06% of the Vicuna-7b model, which corresponds to 5 million parameters. Our experiments primarily focus on the validation of the methodology. For broader applicability, we chose a model that can be trained on a single NVIDIA A100-40G GPU or equivalent, instead of a more powerful but larger model. For simplicity, we used a fixed set of hyperparameters (in Appendix B). Task-specific prompts for both LLM and VLM were designed manually, inspired by prompt templates used by You et al. (2023); Liu et al. (2023).

Table 4 contains all shared hyperparameters from all experiments, including random seed and Lora-related settings. As shown in Table 5, the top group (SL) reports the hyperparameters for both

| Hyperparameter | Value |
|---|---|
| Random Seed | 1 |
| Lora target | q, v |
| Lora r | 8 |
| Lora alpha | 16 |
| Lora dropout | 0.05 |

Table 4: Fixed hyperparameters.

|  | Hyperparameter | LLM | VLM |
|---|---|---|---|
| (SL) | Learning rate | 2e-5 | N/A |
|  | Warmup steps | 100 | N/A |
|  | Truncation | False | N/A |
| (RL) | IM turns | 2 | 2 |
|  | Learning rate | 1.3e-5 | 4e-4 |
|  | KL penalty ($\beta$) | 0.15 | 0.15 |
|  | Max new token | 35 | 10 |
|  | Temperature | 0.15 | N/A |

Table 5: Hyperparameters setting for LLM and VLM.

supervised human-prior fine-tuning and baselines implementation (Vicuna-SL). We keep the VLM fixed during this step. The bottom group (RL) shows the setting during the reinforcement learning stage. Both tasks (VQA and VE) share the same hyperparameter setting.