VISION-LANGUAGE MODEL DIALOG GAMES FOR SELF-IMPROVEMENT

Ksenia Konyushkova, Christos Kaplanis, Serkan Cabi & Misha Denil
Google DeepMind, London
corresponding author: kksenia@google.com

ABSTRACT

The increasing demand for high-quality, diverse training data poses a significant bottleneck in advancing vision-language models (VLMs). This paper presents VLM Dialog Games, a novel and scalable self-improvement framework for VLMs. Our approach leverages self-play between two agents engaged in a goal-oriented play centered around image identification. By filtering for successful game interactions, we automatically curate a high-quality dataset of interleaved images and text. We demonstrate that fine-tuning on this synthetic data leads to performance gains on downstream tasks and generalises across datasets. Moreover, as the improvements in the model lead to better game play, this procedure can be applied iteratively. This work paves the way for self-improving VLMs, with potential applications in various real-world scenarios especially when the high-quality multimodal data is scarce.

1 INTRODUCTION

Large language models (LLMs) have achieved remarkable success by training on vast datasets that now include a significant portion of the Internet (OpenAI et al., 2023; Gemini, 2024). Their performance generally scales with training data size (Kaplan et al., 2020), but acquiring new, high-quality data is increasingly challenging, especially for vision-language models (VLMs), which require carefully curated interleaved image and text data. Recent research (Chen et al., 2024; Bai et al., 2022; Yuan et al., 2024; Huang et al., 2023) indicates that self-improvement techniques can use synthetically generated data to overcome this limitation. We introduce a novel self-improvement method based on goal-oriented play between VLMs. This approach provides a scalable way to iteratively generate high-quality synthetic data, which can be used to fine-tune the model for further performance improvement. By carefully designing the game, we can target specific capabilities and domains for improvement, while the goal-oriented nature ensures the quality of the generated data.

We initiate the process with two VLMs which are assigned the roles of "Describer" and "Guesser" in a variant of reference game (Krauss & Weinheimer, 1964; Das et al., 2017; De Vries et al., 2017; Hakimov et al., 2024) which we call "VLM Dialog Game". Using a set of unlabelled images, we construct a game with one target and several distractor images. The Describer answers questions about the target image, while the Guesser poses targeted questions to disambiguate the target from the distractors (Figure 1). While similar games exist, their primary use has been for human-based data collection or VLM evaluation, and our work demonstrates that this game framework also facilitates VLM iterative self-improvement through goal-oriented self-play.

Thanks to their instruction-following and image-understanding capabilities, the pre-trained VLMs achieve a non-zero success rate in this game. This inherent ability provides a scalable method for generating interleaved image-text data. The initial performance is imperfect: The Describer may provide incorrect answers, and the Guesser may ask irrelevant questions. However, the game's structure allows us to identify successful game instances where the Guesser correctly selects the target image. By filtering for these successful dialogs, we automatically obtain a high-quality dataset of interleaved data. This curated dataset is then used to fine-tune the VLM, improving its proficiency in playing the game and, consequently, its overall image understanding capabilities. The improved model has a higher success rate in the game, and it can be further used to generate a better dataset of interleaved data, enabling further cycles of improvement.

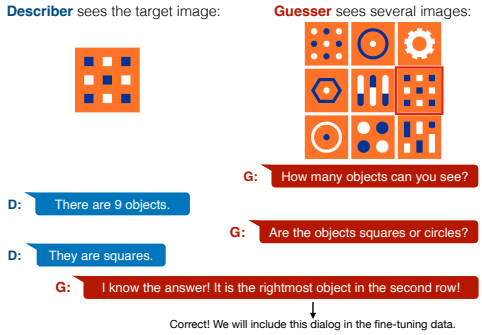


Figure 1: **Example interaction between a Guesser and a Describer in the VLM Dialog Game.** The Guesser aims to identify the target image from a set of distractors by asking questions, which the Describer answers. Since the Guesser correctly identifies the target image at the end of the game, this dialog is considered successful and included in the fine-tuning data.

Our experiments demonstrate that fine-tuning VLMs on the dialog game data yields significant improvements, not just in game performance, but also on related image understanding benchmarks. Fine-tuning on games based on either OpenImages (Kuznetsova et al., 2020) or DOCCI (Onoe et al., 2024) datasets leads to an increase in accuracy on VQAv2 (Goyal et al., 2017) benchmark. The approach is adaptable to specific domains, like robotics, where high-quality data is often scarce. When the game is designed with frames from robotics episodes we significantly improve the model's ability to detect successful execution in robotics tasks.

The remainder of this paper is organized as follows: Section 2 reviews related literature, Section 3 describes the VLM dialog game and self-improvement methodology, Section 4 presents experimental results in general VQA and robotic success detection, and finally Section 5 summarizes findings, discusses limitations, and outlines future research.

2 Related Work

Dialog games Various forms of dialog reference games have been known in linguistics since a long time (Krauss & Weinheimer, 1964). In computer science, prior work on multimodal dialog games is primarily focused on collecting datasets of grounded dialogs (Haber et al., 2019; Das et al., 2017; De Vries et al., 2017; Kazemzadeh et al., 2014) or more recently, evaluating the capabilities of VLMs (Hakimov et al., 2024; Chalamalasetti et al., 2023). These existing games vary in design, including the number of images involved (single or multiple), the roles of the agents (symmetric, sharing the same goal, or asymmetric), and the interaction length (single-turn or multi-turn).

In contrast to these evaluation and data collection efforts, our work leverages dialog games for self-improvement, creating synthetic datasets to enhance VLM capabilities. To the best of our knowledge, this is a novel use of the dialog games.

Self-Improvement Self-improvement (Chen et al., 2024) techniques have gained significant interest in both language and multimodal learning. A prevalent approach involves using an LLM to critique and refine its own responses (Bai et al., 2022; Yuan et al., 2024). For instance, Huang et al. (2023) demonstrate that fine-tuning on self-generated rationale-augmented answers, without ground truth labels, enhances LLM reasoning. Subramaniam et al. (2025) propose a multi-agent framework where diverse responses from a society of LLMs drive iterative fine-tuning and continuous improvement. Zelikman et al. (2022) leverage the question-answer pairs to generate rationales to augment reasoning in order to improve chain-of-thought reasoning.

Self-improvement has also shown promise in enhancing multimodal understanding of VLMs, albeit with fewer existing studies. A prominent technique in VLM self-improvement is cycle consistency, initially developed for image-to-image translation (Zhu et al., 2017). It enforces that a transformation from a source domain to a target domain, and back, yields an output similar to the input. This principle has been successfully extended to the multimodal domain (Li et al., 2023b;a; Sharifzadeh et al., 2024), often exploiting the symmetry between image captioning and text-to-image generation. Cycles such as text1 $\rightarrow$ image $\rightarrow$ text2 or image1 $\rightarrow$ text $\rightarrow$ image2 aim to ensure similarity between the initial and final elements (text1 & text2, or image1 & image2) while producing data for self-improvement. Cycle consistency is particularly valuable when paired text-image data is limited (Li et al., 2023b), but can also facilitate the generation of novel image compositions with abundant data (Sharifzadeh et al., 2024). Another approach to improving VLMs performance is through generating synthetic datasets by eliciting detailed question-answer datasets (Luu et al.,

2024). Furthermore, self-improvement in VLMs is often tailored to specific applications, such as medical imaging, where data acquisition is challenging (Wang et al., 2024).

In contrast to many LLM self-improvement methods that rely on agent-based response generation and critique, we propose a novel self-improvement framework based on games. While inspired by the underlying principle of cycle consistency, our approach deviates from the traditional image-text-image cycle. Instead of direct image generation, we map a target image to a dialog, and then back to the target image through contrastive image selection thus eliminating the need for text-to-image model.

3 METHOD

This section introduces our method for iterative self-improvement through VLM Dialog Games. We first describe the VLM Dialog Game mechanism and its key properties: self-play and goal-oriented nature, which are crucial for self-improvement (Section 3.1). We then detail the complete self-improvement workflow, including game setup, dialog generation and filtering, model finetuning, final evaluation on the target task, and potentially repeating these steps (Section 3.2).

3.1 VLM DIALOG GAME DESCRIPTION

We introduce a VLM Dialog Game which is constructed using unlabelled images and two VLM agents. The first agent, the **Describer**, is presented with a single target image and is instructed to faithfully answer questions about it. The second agent, the **Guesser**, receives a set of N images, including the target image and several distractor images. The **Guesser**'s objective is to identify the target image by posing questions to the **Describer**. The agents' behaviour is controlled by prompting mechanism for VLMs which is described further in Section 3.2.1.

Figure 1 shows an illustrative example of the VLM Dialog Game in action. All images that the Guesser sees contain white and blue objects on an orange background, thus, to identify the target image the Guesser should focus on more specific properties of the images. To disambiguate, the Guesser initiates a series of clarifying questions, such as "How many objects can you see?" and "Are the objects squares or circles?". Once the responses uniquely define the target image (in this case, by pointing to "9 square objects"), the Guesser successfully identifies it. While resembling a classic reference game used for human data collection and VLM evaluation, this specific design features two key elements enabling VLM self-improvement: self-play for data generation and automatic success determination.

Self-play Current VLMs, thanks to their instruction-following capabilities, demonstrate a non-trivial success rate in playing this dialog game (Hakimov et al., 2024). This capability enables a scalable approach to data collection through self-play of two prompted models.

Success Determination We use the Guesser's final selection to determine the success of the dialog: If the selected image matches the target image, the dialog is considered successful and added to the synthetic training data, otherwise it is discarded. This mechanism provides crucial, automatic quality control.

3.2 WORKFLOW

The properties of the VLM Dialog Game enable the following workflow for self-improvement:

- **Game setup**: Configure the dialog game with a designated unlabelled image dataset.
- **Dialog generation**: Generate dialogs via self-play between the VLM agents.
- **Dialog filtering**: Filter generated dialogs based on the success criteria.
- **Model improvement**: Fine-tune the VLM using the filtered dialog data and evaluate its performance on the target task.
- **Repeat the above steps (if needed)**: Repeat dataset generation with an improved version of VLM.

3.2.1 GAME SETUP

This section details the setup of the VLM Dialog Game, including agent instructions and image selection strategies.

Agent instructions We provide precise instructions to both the Describer and Guesser agents to guide their interaction in the game. The Describer is instructed to answer questions about the target image truthfully and accurately. The Guesser agent operates in two stages:

1. Questioning/Guessing: Initially provided with an empty image description, the Guesser must either:
   - Ask a clarifying question to distinguish the target image from the distractors, or
   - Make a guess identifying image X as the target image where X is the index of the hypothesized target image among the distractors.
2. Summarisation: The Guesser must create a concise summary of the target image description given the initial image description (or the previous summary), a question and an answer.

Specific prompt texts for both agents are provided in Appendix A.1.

Image selection and game difficulty The images used in the game can be sourced from various datasets, including general datasets of natural images like OpenImages (Kuznetsova et al., 2020) or DOCCI (Onoe et al., 2024), or domain-specific datasets tailored to applications such as robotics (Zhao et al., 2023). The game's difficulty is controlled through two primary factors related to which images are selected for a game:

- Number of distractors: Increasing the number of distractor images directly increases difficulty. This is due to: (1) the Guesser needing to attend to a larger context, and (2) an increased likelihood of a distractor closely resembling the target image.
- Image similarity in each game: Randomly selecting images from the dataset creates an easier game, while grouping visually or semantically similar images increases the challenge.

We select the appropriate settings in the game so that the games are sufficiently difficult to produce interesting dialogs, but still feasible so that we generate sufficient amount of synthetic training data.

3.2.2 DIALOG GENERATION

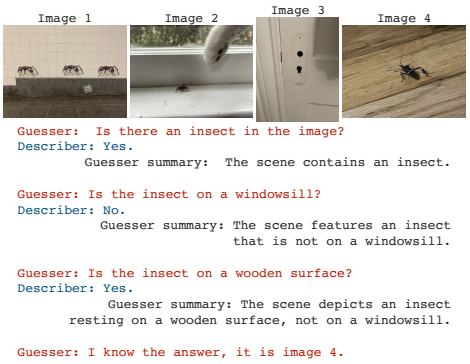
During this stage, the Describer and Guesser agents engage in an interactive dialog. We construct the training dataset from examples of successful behavior by both the Describer and the Guesser:

- **Describer examples.** Input: a single image and a question about it; Output: the corresponding answer.
- **Guesser examples.** Input: N images (including the target and distractors) and a cumulative summary of the target image description; Output: either a clarifying question or a guess identifying the target image.

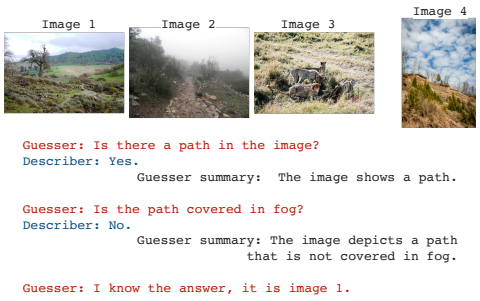
Each successful VLM Dialog Game generates multiple training examples of both types.

3.2.3 DIALOG FILTERING

The game's design allows us to directly verify the Guesser's final selection. However, to mitigate the possibility of correct guess occurring by chance, we perform an additional validation step. We rerun the dialog without the final selection using the same images but in a permuted order and verify that the correct target image is identified in all cases. Empirically, we observed that the position of the *target* image has the most significant impact on the Guesser's accuracy, while the relative order of the distractors (given a fixed dialog) has a smaller effect. Therefore, for computational efficiency we limit the tested permutations to N where we ensure that the target image appears at each possible position (1 to N), while the distractors order can remain fixed. The datapoints from these consistently successful games form the filtered dataset for subsequent model training.



(a) DOCCI



(b) OpenImages

Figure 2: **Examples of VLM Dialog Games** using images from DOCCI and OpenImages datasets. The figure shows the Guesser's questions, the Describer's answers, and the Guesser's internal dialog summary. The Guesser correctly identifies the target image (4 and 1) at the end of the dialog.

3.2.4 Model Improvement

The filtered dataset from successful dialog games is then used to fine-tune the VLM in a standard supervised fine-tuning way. If the gains in playing the VLM Dialog Games are large, we can use the improved model in order to collect the new synthetic dataset for further model improvement. While this process directly affects the VLM's performance within the dialog game itself, our primary focus is on evaluating its capabilities on downstream tasks. For instance, if the dialog game utilizes images from a robotics domain, we assess the fine-tuned VLM's performance on tasks such as robotic success detection, and for general images we test the performance on visual question answering (VQA) on the unseen images.

4 Experiments

4.1 Experimental Setup

We evaluate our method using the Gemini 1.5 Flash model (Gemini, 2024) as the base VLM. Gemini 1.5 Flash is a powerful, instruction-tuned VLM that can take as input interleaved text and images and it provides a strong base model. We use standard supervised fine-tuning procedure (see Appendix D). We limit the game length to a maximum of three question-answer turns. For conciseness, we refer to the self-improvement method of the fine-tuning on synthetically collected dialogs as "VLM Dialog Games".

4.2 Experiments with General Images in Dialog Games

This section details our experiments using the DOCCI (Onoe et al., 2024) and the OpenImages datasets (Kuznetsova et al., 2020) to evaluate the effectiveness of our self-improvement method for image understanding through VQA tasks.

4.2.1 Dataset and Game Configuration

DOCCI dataset contains clusters of images grouped by their category. We randomly sample 1000 image groups, each containing $N = 4$ images from one of 149 categories. Figure 2(a) provides an example of a dialog game generated by prompted Gemini using this setup.

OpenImages We select a subset of 1000 random images, forming them into games with $N = 4$ images. As the dataset does not contain clusters, we select the most similar images (Jia et al., 2021) as distractors. An example of a dialog game produced in this scenario is demonstrated in Figure 2(b).

Table 1: **Comparison of VLM Dialog Games and the base Gemini 1.5 Flash.** Fine-tuning on dialog game data improves both game success rate and VQA performance (yes/no and counting subsets). Results demonstrate generalization across training and evaluation datasets.

Model	game success		VQA	VQA
	DOCCI	OpenImages	yes/no	counting
Gemini 1.5 Flash	20.3%	18.4%	73.0%	56%
VLM Dialog Games (DOCCI)	24.4%	21.9%	79.8% (+6.8)	58.3% (+2.3)
VLM Dialog Games (OpenImages)	25.6%	23.6%	83.4% (+10.4)	56% (+0.0)

4.2.2 Evaluations Tasks

Dialog success rate Following prior work using dialog games to assess VLM capabilities (Hakimov et al., 2024), we use the dialog game success rate as one of measures of the model's improvement. We report the percentage of games where the Guesser correctly identifies the target image across all N tested permutations (as described in Section 3.2.3).

Visual question answering (VQA) To assess the broader impact of our self-improvement method on general visual understanding, we evaluate the fine-tuned model on a subset of 2000 randomly sampled datapoints from the VQAv2 dataset (Goyal et al., 2017). We focus on two specific question types:

- **Binary (yes/no) questions**: Semantically equivalent phrasings (e.g., "No" and "There is no cat") are treated as correct. We report the model accuracy.
- **Object counting questions**: All answers and ground truth labels are converted to numerical form (e.g., "one" becomes "1", "none" becomes "0"). We report a strict exact-match accuracy.

4.2.3 Results

Table 1 compares the performance of the base Gemini 1.5 Flash model with VLM Dialog Games method. Fist, results demonstrate that the VLM Dialog Games method with either the DOCCI or OpenImages datasets improves performance within the game with both training and unseen images (e.g., games played on DOCCI by a model trained with OpenImages). More importantly we also achieve better performance on broader visual understanding tasks as measured by VQA accuracy. Note that evaluation images for it are drawn from a distinct dataset (VQAv2), demonstrating the generalization of our method. Specifically, for DOCCI dialog games, the accuracy on the VQAv2 yes/no and counting subsets increased by 6.8% and 2.3%, respectively. For OpenImages dialog games, yes/no question accuracy increases by 10.4% and remains unchanged for counting questions. We hypothesis that different image sources may be better suited for improving specific tasks. For example, Onoe et al. (2024) note that many DOCCI images contain references to counts, suggesting that this dataset is well-suited for self-improvement on counting task.

4.3 Ablation Studies

Next, we investigate the impact of key design choices: the number of images per game and the method of image grouping. We test the different options on OpenImages dialog games and VQA yes/no question accuracy.

Impact of the number of images per game We study the effect of N on the game complexity by varying N from 2 to 8 (see Appendix B for dialog examples). Table 2 presents the game success rate, the number of question-answer pairs from successful dialogs, and the VQAv2 yes/no accuracy for each N. While fine-tuning with data from any N improves VQAv2 performance compared to the base Gemini 1.5 Flash model, the best result is achieved with $N = 4$ in this study. With $N = 2$, the game is relatively simple, leading to a high success rate but potentially less informative data, and a higher probability of erroneous data due to the correct guesses by chance. Conversely, with $N = 8$, the game becomes too difficult, resulting in few successful dialogs for fine-tuning. These results

Table 2: **Impact of varying the number of images N per game**: We report the number of successful dialog games (out of 1000), the total number of question-answer pairs extracted, and the VQAv2 yes/no accuracy after fine-tuning. The optimal N in this case appears to be 4, balancing game difficulty and data quantity.

N	game success	VQA yes/no
2	83.7%	81.3% (+8.3%)
4	18.4% 275	83.4% (+10.4%)
8	0.24% 34	77.1% (+4.1%)
Gemini 1.5 Flash	73.0%	

Table 3: **Impact of image grouping strategy:** Both random and semantically similar image groupings lead to significant performance gains compared to the baseline. Although using semantically similar images demonstrates slightly better results, the difference is small, highlighting the robustness of the VLM Dialog Game approach even with random image selection.

Image grouping strategy	game success	VQA yes/no
None (initial)	N/A	73.0%
Similar images	18.4%	83.4% (+10.4%)
Random images	24.7%	82.6% (+9.6%)

confirm that balancing game difficulty and the quantity of training data is crucial for generating an optimal dataset for fine-tuning.

Impact of Image Grouping Strategy We investigate how image grouping affects model performance by comparing two strategies: 1) similarity-based grouping (Section 4.2.1), which uses visually and conceptually related distractors to elicit more targeted Guesser questions, and 2) random distractor selection. Table 3 compares models using these strategies. Both strategies improve over the initial Gemini 1.5 Flash checkpoint (73.0%) significantly, therefore, the VLM Dialog Game can be effectively implemented even with random image groupings. However, using similar images yields slightly higher accuracy (83.4% vs. 82.6%). While random images produce a larger quantity of successful dialogs (24.7% vs. 18.4%), the increased challenge of similar images in a game likely leads to more informative training data. Thus, we believe that for the best results in fine-tuning, we need to find a right trade off between game difficulty and training data quantity.

4.4 Robotics Dialog Games

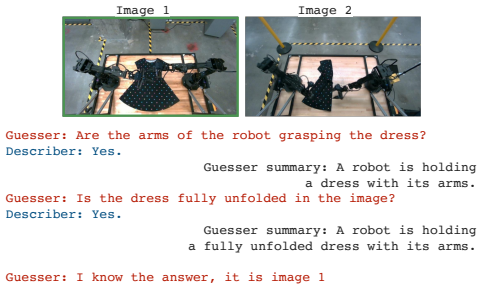


Figure 3: **An example of a dialog game in the robotics domain.** The figure shows the Guesser's questions, the Describer's answers, and the Guesser's internal dialog summary. The Guesser correctly identifies the target image (1) at the end of the dialog.

This section describes our experiments using the VLM Dialog Games on video frames from a robotics manipulation domain where we test VLM success detection in object manipulation tasks. This domain was chosen because high-quality interleaved data is scarce in specialized domains, potentially limiting base model performance in applications.

4.4.1 Dataset and Game Configuration

We use image frames from videos recorded in the ALOHA setup (A Low-cost Open-source Hardware System for Bimanual Teleoperation) (Zhao et al., 2023). The images feature bimanual robotic arms performing 10 object manipulation tasks: 1) fold the dress, 2) put the bowl into the drying rack, 3) unbuckle the belt, 4) open the drawer, 5) put the legos into the lego bag, 6) put the cheese in the basket, 7) remove the gears from the board, 8) put banana into the drying rack, 9) close the green trash bin lid, 10) put the giraffe in the rack (Team, 2025). We use images captured from an overhead camera perspective. Our dataset comprises 20 episodes (both successful and unsuccessful) for each of the 10 tasks, totaling 200 episodes. We limit the game to only two images randomly sampled from the *same task* execution as the success rate drops significantly with more images. We generate 1000 games for each of the 10 tasks by sampling different frame combinations. Figure 3 shows a dialog game example.

Table 4: **Success detection accuracy on the ALOHA dataset**, averaged across 10 tasks. Fine-tuning on dialog game data outperforms the initial checkpoint and the other baselines. Iterative refinement further improves performance.

Model	Game Success	Success Detection Accuracy
Gemini 1.5 Flash	14.39%	56.5%
VLM Dialog Games (round 1)	40.15% (+25.76%)	69.5% (+13.0%)
VLM Dialog Games (round 2)	53.74% (+39.35%)	73.0% (+16.5%)
SFT-Description	N/A	65.0% (+8.5%)
Self-QA	N/A	67.0% (+10.5%)
VLM Dialog Games (answers only)	17.92% (+3.53%)	68% (+12.5)%

4.4.2 Evaluation Task: Success Detection in Robotics

To evaluate the impact of our method on robotic task understanding, we measure the model's ability to perform success detection. Accurate success detection is critical for various robotics applications, including policy training, evaluation, and data curating. We evaluate success detection on the final frame of video episodes, treating it as a zero-shot VQA task (Du et al., 2023). The model is presented with the final frame image and a textual description of the intended task (e.g., "open the drawer") and it is prompted with a question on task completion (e.g., "Is the drawer open?"). We report the accuracy of the model's yes/no responses.

4.4.3 Baselines

To isolate the specific contribution of the VLM Dialog Games, we compare our method against the original Gemini 1.5 Flash model and several other baselines.

Description Supervised Fine-Tuning (SFT-Description) Since our dialog games design utilizes task descriptions for each robotic episode, we include a baseline fine-tuned directly on image-description pairs. This baseline "SFT-Description" helps determine if simply exposing the model to paired image and task descriptions from the target domain is sufficient for improvement.

Self-Improving Question Answering (Self-QA) This baseline explores an alternative self-improvement approach based on question answering similar to the approach of Luu et al. (2024) (without the image captioning). The model performs two tasks:

1. **Question generation:** Given an image from the ALOHA dataset, the model generates a question about the scene.
2. **Answer generation:** Given an image and a generated question, the model provides an answer.

The prompts used for these tasks are detailed in Appendix A.2. This baseline tests whether a simpler self-improvement loop without the goal-oriented dialog structure can achieve similar results.

VLM Dialog Games (Answers Only) Our fine-tuning data includes both Describer and Guesser perspectives. Since the final success detection task closely resembles the Describer's role of answering questions, we include a baseline fine-tuned only on the datapoints from the Describer. This isolates the contribution of the Guesser's questions to the overall improvement.

Multiple Rounds of Self-Improvement We expect fine-tuning to improve the model's performance in subsequent games. Thus, we use the improved model to generates a new, higher-quality dataset of synthetic dialogs. These dialogs are filtered and used to fine-tune the next iteration of the model, a process we refer to as "round 1" and "round 2".

In all cases we generate datasets with a size equivalent to the corresponding dialog game dataset and use it to fine-tune the Gemini 1.5 Flash model with the same settings.

4.4.4 RESULTS

Table 4 presents the success detection accuracy and game success rates averaged across the 10 tasks. The initial Gemini 1.5 Flash model achieves a success detection accuracy of 56.5% on this highly specialised domain, only slightly above chance. Both the SFT-Description and Self-QA baselines improve upon this, demonstrating the benefit of domain-specific fine-tuning (65.0% and 67.0% accuracy, respectively).

However, fine-tuning on a single round of dialog game data (VLM Dialog Games (round 1)) yields a larger improvement, achieving a success detection accuracy of 69.5% surpassing the baseline Self-QA by 2.5%. Interestingly, although we gave no explicit instructions for success detection, the need to distinguish between frames from the *same* task results in questions about the task progression. In contrast, the Self-QA method primarily generates object-related questions (see Appendix C).

Importantly, the round of fine-tuning also substantially increases the game success rate, from 14.39% to 40.15%, thus enabling further improvement. We performed a second round of fine-tuning (VLM Dialog Games (round 2)), using data generated by the round 1 model. This further boosted both the game success rate (to 53.74%) and the success detection accuracy (to 73.0%), a 16.5% absolute improvement over the original base model.

The VLM Dialog Games (answers only) baseline, which uses only the Describer's answers from the dialog games, achieves a success detection accuracy comparable to VLM Dialog Games (round 1). However, its game success rate remains comparatively low (17.92%) and does not enable further iterative improvement. This suggests that the Guesser's questions play a crucial role in improving the model's ability to play the dialog game, which is necessary for continued self-improvement. To conclude, our dialog game framework enables significant adaptation to specialized tasks like robotic success detection, where standard VLM pre-training may be less effective due to the lack of the domain-specific data. Crucially, this self-improvement is achieved with minimal task-specific supervision, requiring only video episodes to guide the dialog generation.

5 DISCUSSION, LIMITATIONS AND CONCLUSION

This paper introduced VLM Dialog Games as a novel self-improvement framework. Our approach leverages goal-oriented self-play between two agents engaged in a reference-style dialog game. By automatically filtering for successful game interactions, we generate a high-quality dataset of interleaved image and text data. Crucially, our approach requires minimal supervision, demonstrating the potential for scalable and data-efficient VLM training.

Despite promising results, our approach has limitations. First, agents might discover trivial or useless "winning" strategies (e.g., querying specific pixel colors or inventing a private vocabulary) without genuine understanding. To avoid these trivial strategies, we used several properties of the game: 1) we rely on the instruction-following capabilities of the base model to steer the game behaviour, 2) our image grouping strategy makes it less likely that trivial questions would help to distinguish images. As we see the improvement on the downstream tasks, we conclude that we managed to largely avoid trivial strategies, but developing automated and robust methods for detecting and preventing such behavior remains an important area for future research. Second, while we saw significant improvement in robotics (likely due to its under-representation in pre-training), iterative gains plateaued after 2 iterations. It seems that when the success rate on the game is not changing by a large margin, the difference in the size of the fine-tuning dataset is not large enough to show the difference in the final performance. This is a known property of self-improvement methods and further research into techniques that can mitigate this plateauing effect in VLM dialog games and other self-improvement methods would be a valuable direction for future work. Finally, model improvement in the model comes at a computational cost to produce data with model inference and cost to fine-tune the model and other cheaper strategies can be considered, such as prompt engineering and inference-time computation.

We believe that the success of our method suggests a promising direction for model development. While current VLMs continue to improve, the core principle of our approach – learning from successful interactions in a goal-oriented setting – remains applicable. Multimodal dialog games offer a general recipe for VLM self-improvement, adaptable to various domains and tasks, particularly those with scarce or specialized data.

REFERENCES

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

Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. clembench: Using game play to evaluate chat-optimized language models as conversational agents. In *EMNLP*, 2023.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *ICML*, 2024.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *CVPR*, 2017.

Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. Guesswhat?! visual object discovery through multi-modal dialogue. In *CVPR*, 2017.

Yuqing Du, Ksenia Konyushkova, Misha Denil, Akhil Raju, Jessica Landon, Felix Hill, Nando de Freitas, and Serkan Cabi. Vision-language models as success detectors. In *Proceedings of The 2nd Conference on Lifelong Learning Agents*, 2023.

Team Gemini. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv:2403.05530*, 2024.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.

Janosch Haber, Tim Baumgärtner, Ece Takmaz, Lieke Gelderloos, Elia Bruni, and Raquel Fernández. The PhotoBook dataset: Building common ground through visually-grounded dialogue. In *ACL*, 2019.

Sherzod Hakimov, Yerkezhan Abdullayeva, Kushal Koshti, Antonia Schmidt, Yan Weiser, Anne Beyer, and David Schlangen. Using game play to investigate multimodal and conversational grounding in large multimodal models. *arXiv:2406.14035*, 2024.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In *EMNLP*, 2023.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

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

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.

RM Krauss and S Weinheimer. Changes in reference phrases as a function of frequency of usage in social interactions. *Psychonomic Science*, 1964.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 2020.

Hang Li, Jindong Gu, Rajat Koner, Sahand Sharifzadeh, and Volker Tresp. Do DALL-E and Flamingo understand each other? In *ICCV*, 2023a.

Tianhong Li, Sangnie Bhardwaj, Yonglong Tian, Han Zhang, Jarred Barber, Dina Katabi, Guillaume Lajoie, Huiwen Chang, and Dilip Krishnan. Leveraging unpaired data for vision-language generative models via cycle consistency. *arXiv 2310.03734*, 2023b.

Duc-Tuan Luu, Viet-Tuan Le, and Duc Minh Vo. Questioning, answering, and captioning for zero-shot detailed image caption. In *Proceedings of the Asian Conference on Computer Vision*, 2024.

Yasumasa Onoe, Sunayana Rane, Zachary Berger, Yonatan Bitton, Jaemin Cho, Roopal Garg, Alexander Ku, Zarana Parekh, Jordi Pont-Tuset, Garrett Tanzer, Su Wang, and Jason Baldridge. DOCCI: Descriptions of Connected and Contrasting Images. In *ECCV*, 2024.

OpenAI, Josh Achiam, 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. *arXiv:2303.08774*, 2023.

Sahand Sharifzadeh, Christos Kaplanis, Shreya Pathak, Dharshan Kumaran, Anastasija Ilic, Jovana Mitrovic, Charles Blundell, and Andrea Banino. Synth ^{2}: Boosting visual-language models with synthetic captions and image embeddings. *arXiv:2403.07750*, 2024.

Vighnesh Subramaniam, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, Shuang Li, and Igor Mordatch. Multiagent finetuning: Self improvement with diverse reasoning chains. *arXiv:2501.05707*, 2025.

Gemini Robotics Team. Gemini robotics: Bringing ai into the physical world. *arXiv:2503.20020*, 2025.

Jinzhuo Wang, Kai Wang, Yunfang Yu, Yuxing Lu, Wenchao Xiao, Zhuo Sun, Fei Liu, Zixing Zou, Yuanxu Gao, Lei Yang, et al. Self-improving generative foundation model for synthetic medical image generation and clinical applications. *Nature Medicine*, 2024.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. Self-rewarding language models. In *ICML*, 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. In *NeurIPS*, 2022.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *RSS*, 2023.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

A Appendix: VLM Prompts

A.1 Prompt for VLM Dialog Games

Prompt to Describer agent to answer questions about the image faithfully is the same for all datasets and domains:

```
You are given an image and your task is to answer a given question about
    it. Be precise and accurate. Only answer the question, do not say
    anything else about the image.
Image: {image}
Question: {question}
Answer:
```

Prompt for Guesser for general images (could be adapted for various number of images):

```
You are given several images Image 1, Image 2, Image 3, Image 4 and image
     description.
This image description refers to only a single image, however, the image
    description might be incomplete.
You task is the following:
```

```
1) If the image description can only refer to a single image from the set
     of images (Image 1, Image 2, Image 3, Image 4) you should provide
    the answer in the format:
Answer: I know the answer, it is image X.
where X is the index of an image (1, 2, 3, 4).
Only provide a response in this format when you are absolutely certain to
     which image the image description refers to.
Never provide an answer in this format when the image description is
    empty.
2) If no image description is provided or the image description can refer
     to more than one image, your task is to ask additional question to
    narrow down the space of possible images from the set (Image 1, Image
     2, Image 3, Image 4).
Ask any question that would help you to narrow the space of possible
    images.
Choose a question that would help you to maximise the information about
    the content of the target image.
Try to ask objective, factual questions that cover the content of the
    image, but not the deductions about the scene or any impressions
    about the image.
Follow the format:
Question: put your question here.
So, now given the image descriptions and 4 images, decide if you are
    going to make a guess (in that case produce an Answer) or ask a
    question (in that case produce a Question).
Image description: {image_description}
Image 1: {image1} Image 2: {image2} Image 3: {image3} Image 4: {image4}
```

In robotics experiments, the Guesser agent was prompted with instructions very similar to the general images, but mentioning that the images come from a robotics domain. These instructions also direct the agents to focus on relevant visual features for robotic manipulation tasks as opposed to details in the background (e.g., people or chairs):

```
You are given two images (Image 1, Image 2) from a scene where robot is
    trying to {task} and image description.
This image description refers to only a single image, however, the image
    description might be incomplete.
You task is the following:
1) If the image description can only refer to a single image from the set
     of images (Image 1, Image 2) you should provide the answer in the
    format:
Answer: I know the answer, it is image X.
where X is the index of an image (1,2).
Only provide an response in this format when you are absolutely certain
    to which image the image description refers.
Never provide an answer in this format when the image description is
    empty.
2) If no image description is provided or the image description can refer
     to more than one image, your task is to ask additional question to
    narrow down the space of possible images from the set (Image 1, Image
     2).
Try to ask objective, factual questions that cover the content of the
    image.
Choose a question that would help you to maximise the information about
    the content of the image.
NEVER ask questions about the background of the robotic scene (e.g.,
    people in the background, scooters or chairs).
NEVER ask questions about the facts that are already known from the image
     description.
Follow the format:
Question: put your question here

So, now given the image descriptions and 2 images, decide if you are
    going to make a guess (in that case produce an Answer) or ask a
    question (in that case produce a Question).
```

```
Image description: {image_description}
Image 1: {image1} Image 2: {image2}
```

Prompt to Guesser agent to summarise the state of the existing dialog for all domains:

```
You are given a short description of a scene and one question and answer
    about it.
Your task is to summarise the content of the scene in a short sentence or
     paragraph. Only provide a summary, do no output anything else.
Always include all the details 1) from the description, 2) from question-
    answer pair into your summary.
Description: {description}
Question: {question}
Answer: {answer}
Your summary:
```

A.2 Full Prompts for the Baseline Self-QA

To ask questions:

```
You are given an image and your task is to ask a question about the
    content of this image.
Try to ask objective, factual questions that cover the content of the
    image, but not the deductions about the scene or any impressions
    about the image.
NEVER ask questions about the background of the robotic scene (e.g.,
    people in the background, scooters or chairs).
Follow the format:
Question: put your question here.
So, now given the image, ask a question.
Image: {image}
Question:
```

To answer the questions:

```
You are given an image and your task is to answer a given question about
    it. Be precise and accurate. Only answer the question, do not say
    anything else about the image.
Image: {image}
Question: {question}
Answer:
```

B Appendix: Dialog Examples for Varying Number of Images

Figures 4, 5, and 6 illustrate dialog examples with a consistent target image (Image 1) but a varying number of distractor images (1, 3, and 7, respectively, corresponding to N values of 2, 4, and 8). These examples demonstrate the effect of N on dialog length and complexity. With only two images ($N = 2$), the dialog is short, focusing on a single distinguishing feature. With four images ($N = 4$), the dialog becomes more complex, requiring two questions that progressively narrow down the possibilities. However, with eight images ($N = 8$), the Guesser is unable to identify the target image within the three-question limit.

C Appendix: Question-Answers Generated by Dialog Games and Self-QA

Self-QA:

- Question: Is there a yellow object in the image? Answer: yes
- Question: Is there a red object on the surface? Answer: yes

Figure 4: An example of a dialog game with two images.

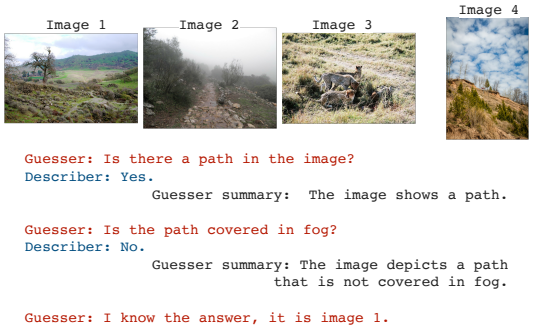


Figure 5: An example of a dialog game with four images.

- Question: Are there any balls inside the basket? Answer: No
- Question: Is there a red ball in the image? Answer: yes
- Question: Are there two robotic arms in the image? Answer: yes
- Question: Is there a red apple in the image? Answer: yes
- Question: Is there a red object on the floor? Answer: yes
- Question: Is there a piece of fruit in the basket? Answer: yes
- Question: Is there a yellow triangle in the image? Answer: yes

VLM Dialog Games:

- Question: Are there any Lego blocks on the floor that are not in the bag? Answer: Yes.
- Question: Is the drawer in the image open? Answer: Yes.
- Question: Is the trash bin lid open or closed? Answer: Closed.
- Question: Is the bowl inside the drying rack? Answer: No.
- Question: Is the cheese in the basket? Answer: No.
- Question: Is the banana inside the drying rack? Answer: Yes.
- Question: Is the robot's gripper holding the belt? Answer: No.
- Question: Is there a basket in the image? Answer: No.
- Question: Is the drawer open? Answer: Yes.

D APPENDIX: LLM INFERENCE AND TRAINING DETAILS

We rely on Gemini 1.5 Flash (gemini-1.5-flash-002) model which is available for inference and fine-tuning through the Google Cloud Vertex API. For generating dailog and evaluation, we sample with nucleus sampling selecting the top 0.8 probability mass of tokens. For the evaluation, we use sampling temperature 0. Our batch size for SFT is 16, we use Adam optimizer with learning rate 5×10^{-7}. To prevent overfitting to the small datasets from the dialog games, we use a small

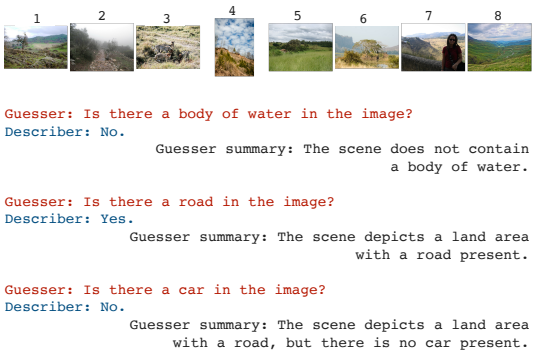


Figure 6: An example of a dialog game with eight images.

unrelated to the any of the tested tasks dataset of images with text and track token loss on it. We select a checkpoint just before this loss starts increasing. This usually corresponds to approximately one epoch of fine-tuning. In various baselines, we generate datasets with a size equivalent to the corresponding dialog game dataset and use it to fine-tune the Gemini 1.5 Flash model with the same settings.