Can Feedback Enhance Semantic Grounding in Large Vision-Language Models?

Yuan-Hong Liao ANDREW@CS.TORONTO.EDU
University of Toronto & Vector Institute

Rafid Mahmood MAHMOOD@TELFER.UOTTAWA.CA
University of Ottawa & NVIDIA

Sanja Fidler FIDLER@CS.TORONTO.EDU
University of Toronto & Vector Institute & NVIDIA

David Acuna DACUNAMARRER@NVIDIA.COM
NVIDIA

Abstract

Enhancing semantic grounding abilities in Vision-Language Models (VLMs) often involves collecting domain-specific training data, refining the network architectures, or modifying the training recipes. In this work, we venture into an orthogonal direction and explore whether VLMs can improve their semantic grounding by 'receiving' feedback, without requiring in-domain data, fine-tuning, or modifications to the network architectures. We systematically analyze this hypothesis using a feedback mechanism composed of a binary signal. We find that if prompted appropriately, VLMs can utilize feedback both in a single step and iteratively, showcasing the potential of feedback as an alternative technique to improve grounding in internet-scale VLMs. Furthermore, VLMs, like LLMs, struggle to self-correct errors out-of-the-box. However, we find that this issue can be mitigated via a binary verification mechanism. Finally, we explore the potential and limitations of amalgamating these findings and applying them iteratively to automatically enhance VLMs' grounding performance, showing grounding accuracy consistently improves using automated feedback across all models in all settings investigated. Overall, our iterative framework improves semantic grounding in VLMs by more than 15 accuracy points under noise-free feedback and up to 5 accuracy points under a simple automated binary verification mechanism.[1]

Keywords: Vision-Language Models, Visual Grounding, Prompt Engineering, Feedback

1 Introduction

The evolution of Large Language Models (LLMs) to encompass multimodal inputs has given rise to an emerging paradigm of general-purpose models that can solve multimodal understanding problems via user-prompt interaction (Touvron et al., 2023; Team et al., 2024, 2023; Yang et al., 2023c; McKinzie et al., 2024). Vision-Language Models (VLMs) are a growing family of multimodal models that simultaneously understand both visual and

1. Project website is hosted at `https://andrewliao11.github.io/vlms_feedback/` and code is released at `https://github.com/andrewliao11/vlms_feedback`

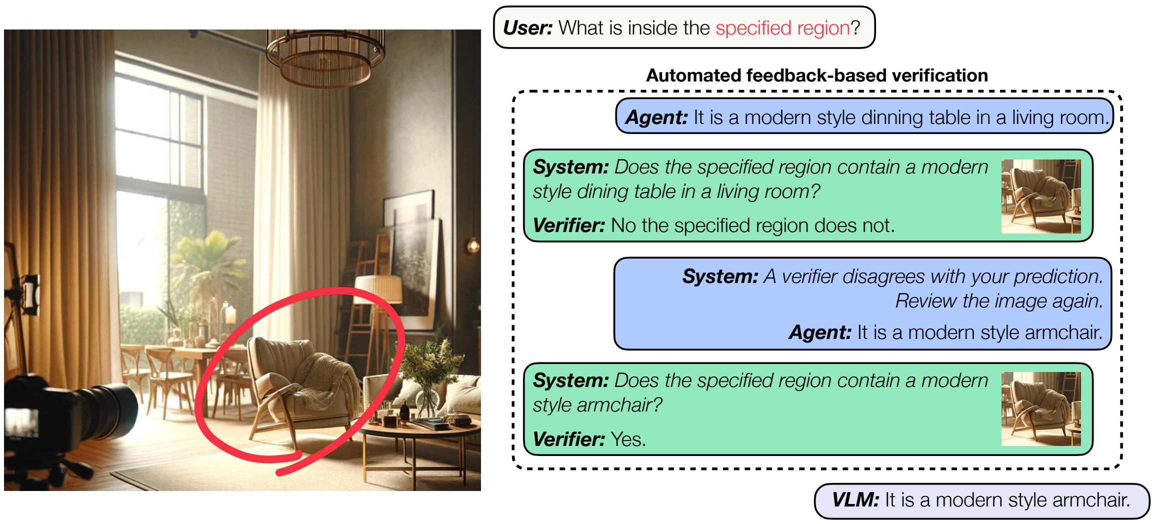

Figure 1: **Enhancing semantic grounding in VLMs with feedback.** We investigate whether VLMs can improve their grounding performance through feedback, without requiring in-domain data, fine-tuning, or modifications to the network architecture. Given an image and a specified region, a VLM is asked to identify the semantic properties that best describes the content in that region. Behind the scenes, an automatic feedback-based verification mechanism manages an interaction between the VLM and a 'Verifier' to refine the VLM's initial understanding.

language input cue. These models have demonstrated strong zero-shot performance on tasks including image classification (Deng et al., 2009), captioning (Young et al., 2014), visual question answering (Antol et al., 2015; Goyal et al., 2017), and commonsense reasoning from image data (Yu et al., 2016; Yuksekgonul et al., 2023), motivating application-specific VLMs in autonomous vehicles and robotics (Cui et al., 2024; Nasiriany et al., 2024b).

Despite VLMs' strong visual-language understanding abilities, fine-grained visual grounding remains a challenge. Specifically, VLMs struggle to understand region-specific information within complex scenes, for example, when the models are prompted to describe specific objects within a crowded image (Chen et al., 2023; Yang et al., 2023a; You et al., 2023). This limitation stems from various factors, such as the limited spatial awareness of the vision encoder (Guo et al., 2024; Li et al., 2024, 2023), insufficient or inadequate training data (Liu et al., 2024), sub-optimal training recipes (Lin et al., 2023), and sensitivity to prompt design (Yang et al., 2023b). While these factors can be specifically addressed with improved training recipes, these methods demand considerable computational cost (Cai et al., 2023; You et al., 2023; Chen et al., 2023; Yang et al., 2023b). This makes the adaptation of VLMs for fine-grained visual grounding tasks prohibitive to practitioners that do not have the resources needed to retrain these models or that rely on API-based VLMs for their downstream applications.

The adjacent LLM literature has demonstrated solving complex tasks via automated feedback from application-specific systems or even other LLM agents (Peng et al., 2023;

Madaan et al., 2024; Du et al., 2024). For example, self-debugging (Chen et al., 2024) can identify and fix errors in code generation by simply parsing the execution results. Feedback mechanisms act as verifiers which review the output of the target model and provide a signal that can be prompted back to the target model (Chen et al., 2024; Gou et al., 2024). This interaction allows LLMs to iteratively generate high-quality outputs for difficult tasks without any fine-tuning (Zhang et al., 2023a). Furthermore, automated feedback-based systems can be developed with readily-available models using only API calls, which reduces the barriers for practice. Most importantly, VLMs have already demonstrated the ability to verify some visual tasks such as evaluating image caption pairs (Lu et al., 2024; Zhang et al., 2023b). Consequently, we believe the lessons learned from the literature on automated feedback for LLMs should be applicable to fine-grained visual grounding for VLMs, thereby unlocking performance gains on complex multimodal tasks without expensive retraining, data, or model architectural changes.

In this paper, we perform the first exploration of whether VLMs can improve their semantic grounding on zero-shot test tasks by 'reading' automated prompt-based feedback, as shown in Fig. 1. We analyze state-of-the-art open-source VLMs including LLaVA-1.5 (Liu et al., 2023a), ViP-LLaVA (Cai et al., 2024), CogVLM (Wang et al., 2024), and a proprietary VLM, *i.e.*, GPT-4V (Yang et al., 2023c), to reveal consistent trends. Following recent work (Yang et al., 2023a; Zhang et al., 2024), we repurpose the challenging panoptic segmentation datasets in ADE20k (Zhou et al., 2017) and COCO (Lin et al., 2014) for semantic grounding. We anticipate this analysis also translates to other complex tasks. We first examine whether feedback can enhance semantic grounding in VLMs and more specifically, what type of feedback is necessary for the VLMs to improve their downstream task performances. We systematically evaluate noise-free verification signals from an oracle and a diverse range of feedback mechanisms including language and visual prompting. We then examine whether a VLM can verify their answers to complex visual grounding tasks and what type of feedback can a VLM-based verifier generate with high-quality. Finally, we close the loop between receiving and sending feedback to propose an automated iterative feedback-based mechanism that yields VLMs up to nearly 5 accuracy points improvements on semantic grounding by essentially trading performance with compute. Our key findings are:

1. **VLMs can read the feedback to improve downstream semantic grounding.** In a single-step setting with noise-free binary signals, VLMs improve their semantic grounding performance by 4 to 12 accuracy points, and over multiple rounds, by over 15 points across five rounds. This shows the potential of feedback as a means of improving grounding performance in VLMs. Perhaps surprisingly, providing the explicit class label as feedback does not automatically improve accuracy to 100%, identifying a limitation in open-source VLMs for handling tasks relying purely on language understanding.

2. **VLMs can be used as binary feedback providers.** Although, similar to LLMs (Huang et al., 2023a; Xu et al., 2024), VLMs struggle to correct themselves out-of-the-box, we show that this issue can be mitigated via a binary verification mechanism that modifies the input image through isolation or marking of objects. Comparing our

binary verification mechanism to *intrinsic self-correction*, we observe up to 18 points improvement in F_1 score.

3. **VLMs benefit from automatic iterative feedback by improving semantic grounding accuracy up to nearly 5 accuracy points.** This is in sharp contrast with an application of prior intrinsic self-correction (Madaan et al., 2023; Kim et al., 2023), which can decrease performance by up to -10.18 accuracy points. Interestingly, our improvements come mostly from the first iteration and in most cases saturates after five iterations. This opens further questions on the quest for better automated verification protocols for semantic grounding. We emphasize our focus is not to develop a SoTA semantic grounding technique or an optimal automatic verification mechanism, but to study the limits and potentials of iterative feedback for semantic grounding. We anticipate future generations of VLMs will yield substantial quantitative advancements in the task at hand.

2 Related Works

2.1 Prompting in LLMs and VLMs

In-context learning in LLMs (Brown et al., 2020) has encouraged various new prompting techniques such as Chain-of-Thought (CoT) (Wei et al., 2022), Least-to-Most (Zhou et al., 2023), and StepBack (Zheng et al., 2024) to improve LLMs' reasoning capability. Chain-of-thought notably demonstrates several reasoning paths for LLMs to encourage them to solve complex task by following similar reasoning paths (Yao et al., 2023; Wang et al., 2023). However, these techniques may be less applicable to VLMs due to the limited in-context learning capability of VLMs, especially the visual instruction tuned VLMs (Zhao et al., 2024; Zeng et al., 2024). In contrast, Zero-shot CoT can elicit model reasoning without the reliance on in-context learning, but instead by simply prepending a guiding sentence to the model responses (Kojima et al., 2022). Moreover, prompting techniques for VLMs have mostly focused on visual prompting. Prior work has shown that models trained on web-scale data can focus on specific visual markers, *e.g.*, red circles (Shtedritski et al., 2023). More recently, Set-of-Marks (SoM) prompting shows that the GPT-4V model is able to ground multiple objects at the same time by simply overlaying a set of object identifiers on the image (Yang et al., 2023a; Nasiriany et al., 2024a). In our work, we leverage these techniques to introduce feedback to VLMs and improve semantic grounding.

2.2 Multimodal Evaluation and Verification

Recent large-scale VLMs such as CLIP (Radford et al., 2021) and GPT-4V (Yang et al., 2023c) have introduced a new paradigm for challenging multimodal evaluation. For instance, evaluating image captions is difficult to do with classical discriminative metrics (Kilickaya et al., 2017; Cui et al., 2018). To provide evaluations that align with human judgements, CLIPScore (Hessel et al., 2021) leverages web-scale VLMs to measure similarity between images and captions. On the other hand, LLMScore (Lu et al., 2023) shows that by coupling an image captioner and an off-the-shelf object detector, LLMs can directly measure the alignment for text-to-image models. More recently, GPT-4V has been used directly as an automatic evaluator for vision language tasks such as text-to-3D generation (Zhang et al.,

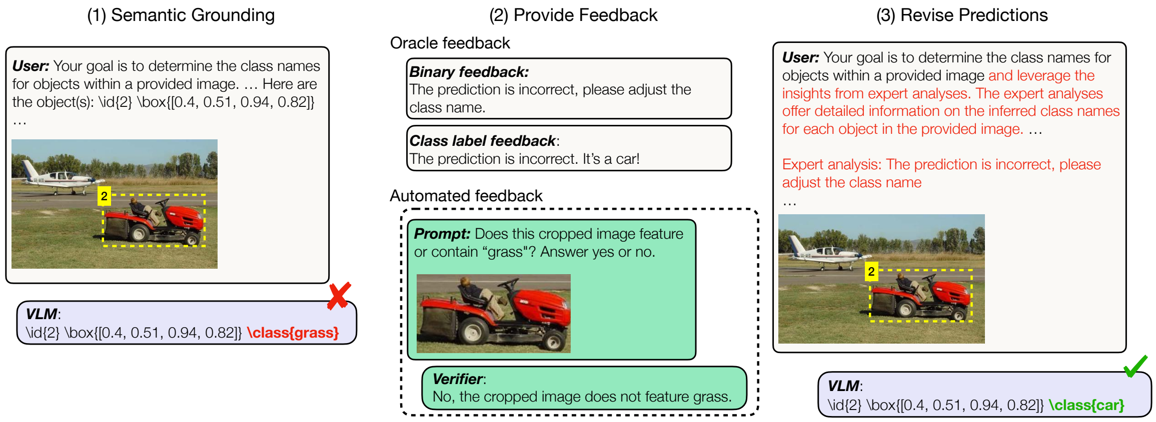

Figure 2: **Feedback Dynamics.** Left (Semantic Grounding): This figure shows the task of semantic grounding. Given an image and a text prompt that specifies a region of interest, a VLM is tasked to identify the semantic class best describing that region. Center (Feedback Mechanisms Explored): We illustrate the various feedback mechanisms examined in this study. *Noise-Free Binary Feedback*: In this setting, the feedback is provided by an oracle that solely addresses the correctness of the class predicted by the VLM. *Noise-Free Class Label Feedback*, the feedback is also provided by an oracle and explicitly provides the correct class label. *Automated Binary Verification*: In this setting, a VLM serves as a verifier, analyzing a modified version of the input image where the region of interest has been isolated. Its task is to conduct a binary verification of the previous prediction. Right (Feedback Receiving): The VLM receives the provided feedback and is then prompted to re-evaluate and adjust the previous prediction.

2023c; Wu et al., 2024). Inspired by the promise of using large VLMs as evaluators, we explore whether VLMs are able to evaluate and verify their own predictions. In contrast to prior work, we treat the same VLMs as both predictors and evaluators/verifiers.

2.3 Self-correction in LLMs using Feedback

LLMs have demonstrated some ability to criticize and refine their own text responses, thereby trading compute for performance (Madaan et al., 2023; Kim et al., 2023). However, a systematic evaluation of these 'self-correction' abilities has shown that the current generation of LLMs can only reliably correct their responses if given access to external techniques (Huang et al., 2023a; Tyen et al., 2024). For example, self-debugging (Chen et al., 2024) improves code generation by interacting with a code executor, whereas the CRITIC framework uses external tools to cross-check and refine initial responses (Gou et al., 2024). This paper presents the first exploration of self-correction for enhancing visual grounding in VLMs. We first verify if VLMs suffer the similar limitation of self-correction and then investigate whether visual prompting techniques can mitigate this implicit self-bias.

3 Feedback Dynamics in VLMs for Semantic Grounding

In this section, we first define the visual grounding problem for VLMs (Sec. 3.1). We then introduce our key research questions on whether VLMs can improve their performances in semantic grounding by receiving prompt-based feedback and whether they can provide feedback to their own semantic grounding predictions (Sec. 3.2). Finally, we summarize the evaluation metrics, datasets, and models comprising our experiment protocol (Sec. 3.3).

3.1 Visual Semantic Grounding

We consider a visual grounding problem that maps a given image region to the text space, referred to as semantic grounding (Zhang et al., 2024; Yang et al., 2023a). Prior work (Lee et al., 2024) has shown that such grounding abilities strongly correlate with the visual reasoning abilities in VLMs. Consider an image $\mathrm{x} \in \mathbb{R}^{\mathrm{h}\times\mathrm{w}\times3}$ where h and w denote the image's height and width, respectively. There exists a priori an image partition function that takes an image and produces N semantically distinct regions $\{\mathrm{r_i}\}_{\mathrm{i}=1}^{\mathrm{N}}$, where each $\mathrm{r_i} \in [0,1]^{\mathrm{h}\times\mathrm{w}}$. A general-purpose VLM (*e.g.*, LLaVA (Liu et al., 2023b)) is then tasked to take the image x, the image region $\mathrm{r_i}$, a text prompt q, and to output text $\mathrm{o_i} = \mathrm{VLM}(\mathrm{x}, \mathrm{r_i}, \mathrm{q})$ that best describes the image region. The output format depends on the evaluation metrics of interest. Fig 2 shows an example of the specific task prompt used in this problem.

Following prior work (Yang et al., 2023a), we use ground truth segmentation masks as semantically distinct image regions $\{\mathrm{r_i}\}_{\mathrm{i}=1}^{\mathrm{N}}$. We evaluate semantic grounding ability by whether the VLM can estimate the ground truth class label for each region in every scene.

3.2 Enhancing Semantic Grounding in VLMs with Noise-Free Feedback

Recently, LLMs have demonstrated significant improvements in performance on complex language semantic tasks such as coding and math reasoning by leveraging sequential feedback from prompts (Chen et al., 2024; Nathani et al., 2023). We note that VLMs can process diverse visual and text inputs while simultaneously sustaining a dialogue from multiple input rounds similar to LLMs. To determine whether VLMs behave similarly to LLMs in improving performance on visual semantic tasks, we direct two series of questions under a noise-free feedback setup: (i) whether VLMs can receive dialogue feedback in order to improve semantic grounding; and (ii) whether VLMs can generate binary grounding feedback for themselves.

3.2.1 Can VLMs Receive Grounding Feedback?

For an image and an image region, a VLM makes an initial prediction without feedback $\mathrm{o_{i,0}} = \mathrm{VLM}(\mathrm{x}, \mathrm{r_i}, \mathrm{q})$, which we refer to as a *base prediction*. Here, we adopt a noise-free setting where an oracle f^* that knows the ground truth class labels for any scene evaluates $\mathrm{o_{i,0}}$ and generates a feedback $f^*_{\mathrm{i},0}$ signal describing potential errors in the base prediction. The feedback can be converted into text or visual marks for the textual and visual prompts, $\mathrm{x}_{|f^*_{\mathrm{i},0}}$ and $\mathrm{q}_{|f^*_{\mathrm{i},0}}$. The enhanced predictions by feedback are represented as $\mathrm{o_{i,1}} = \mathrm{VLM}([\mathrm{x}, \mathrm{x}_{|f^*_{\mathrm{i},0}}], \mathrm{r_i}, [\mathrm{q}, \mathrm{q}_{|f^*_{i,0}}])$, where we overload the operator $[\cdot]$ as overlaying for images and concatenation for text. Fig. 2 (center and right) shows the examples of semantic

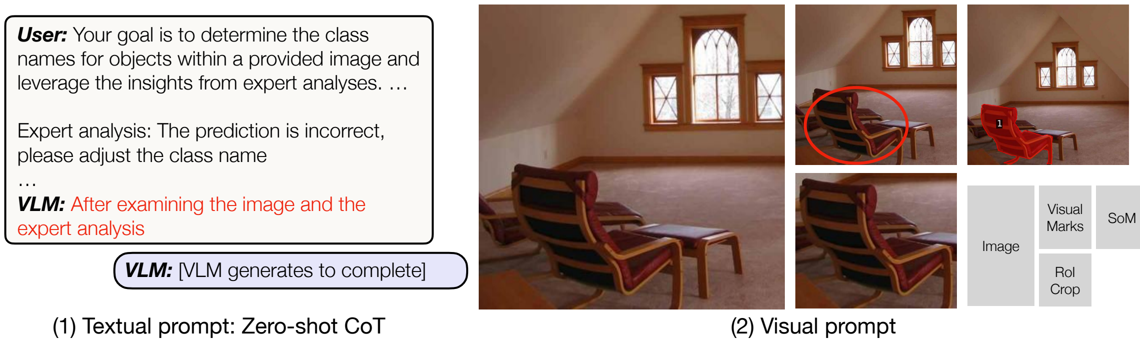

Figure 3: **Examples of textual and visual prompting techniques.** Left: Zero-shot CoT prepends a guiding sentence (in red) before VLMs' output to guide the model. Right: We apply various visual prompting techniques including RoI crop, visual marks, and SoM to modify input images to VLMs to guide the models' attention.

grounding in VLMs with feedback. See the full prompt in Appendix A. Then, we ask the following questions:

Q1 *What type of feedback yields the best improvement in grounding performance?* We consider two alternatives: (i) directly providing the ground truth class labels in a text prompt; and (ii) providing a message on whether the previous prediction is correct (see Figure 2 right).

Q2 *How should the feedback be prompted to a VLM?* We consider several alternatives and their combinations:

(i) *Zero-shot Chain-of-Thought (CoT)*: Kojima et al. (2022) shows that simply prepending a guiding sentence *'Let's think step-by-step'* before generation can strongly guide the LLMs for desired tasks. We adopt a similar idea and use the guiding sentence *'After examining the image and the expert analyses, the final answer is'* for the semantic grounding tasks. Here, the feedback is referred as expert analyses to encourage the model to follow the feedback.

(ii) *Visual Marks*: Shtedritski et al. (2023) shows that Internet-scale vision-language encoders are biased to attend to visual marks (*e.g.*, red circles).

(iii) *Set-of-Mark (SoM)*: Yang et al. (2023a) shows that overlaying object identifiers on the image improves visual grounding in GPT-4V.

To measure improvement, we evaluate the overall improvement in accuracy between the initial and the revised predictions. We show the examples of each textual and visual prompting technique in Fig. 3.

3.2.2 Can VLMs Give Binary Grounding Feedback?

We explore whether a VLM can itself provide feedback. We consider simple binary feedback that evaluates the correctness of the grounding predictions. We refer this VLM to as a 'Verifier' that generates binary feedback from the image, the image region, and the grounding prediction $f^{\mathrm{VLM}}(\mathrm{x}, \mathrm{r_i}, \mathrm{o_i})$, as shown in Fig. 2 center.

While prior work (Kim et al., 2023; Madaan et al., 2023) shows the promise of LLMs to perform self-correction by simply prompting LLMs to review their own responses, a recent study has demonstrated LLMs have limited *intrinsic self-correction* abilities (Huang et al., 2023a). Furthermore, the study suggests that LLMs require *external* techniques to consistently improve their outputs. Motivated by this observation, we ask the following questions for VLMs:

Q3 *Can a VLM generate high-quality feedback via intrinsic self-correction?* We verify whether the prior results demonstrating the challenges of intrinsic self-correction in LLMs (Huang et al., 2023a) hold for VLMs.

Q4 *Can a VLM leverage external techniques,* e.g.*, visual prompting, to overcome the issues in intrinsic self-correction?* We consider several techniques:

(i) *Visual marks:* The verifier receives the image with a highlighted object of interest and a prompt to determine if the predicted class label accurately describes the object (Shtedritski et al., 2023).

(ii) *RoI crop:* Prior work (Gu et al., 2022) distills features of cropped regions to the object detectors. Inspired by this, we design the verifier to receive a cropped image isolating the object of interest.

(iii) A combination of Visual Marks and RoI crop.

For intrinsic self-correction, we adapt prior work in LLMs self-correction (Kim et al., 2023) to semantic grounding task. Here, we prompt the verifier to *'Carefully review and refine your answer'* to automatically correct grounding predictions. Note that although intrinsic self-correction does not explicitly generate a binary feedback, we can obtain the corresponding binary signal by comparing the alignment of the grounding predictions before and after correction.

3.3 Experiment Protocols

Datasets. We use the panoptic segmentation dataset in ADE20k (Zhou et al., 2017) for our analysis. This dataset was not previously used in the instruction tuning of the VLMs that we study, which allows us to better characterize in-the-wild generalization abilities. Further, this dataset contains a validation set of 2k complex, crowded scenes with over 30k masks, labeled across a fine-grained class label spectrum of 150 distinct categories. We validate our results on the iterative setting in COCO panoptic segmentation (Kirillov et al., 2019; Lin et al., 2014). Although the COCO dataset is standard in the visual grounding community, most VLMs are trained on the visual instruction dataset derived from the COCO labels. We, therefore, consider it as an in-domain dataset for most VLMs, in contrast to ADE20k.

	Zero-shot CoT	Visual Prompt	LLaVA-1.5	ViP-LLaVA	CogVLM
Base Predictions	No	No	35.86	35.86	15.98
+ Class Label Feedback	No	No	$94.80_{+58.94}$	$74.99_{+39.13}$	$77.04_{+61.06}$
+ Binary Feedback	No	No	41.04	40.36	16.25
	Yes	No	43.30	42.00	18.25
	Yes	SoM	42.41	44.53	18.64
	Yes	Visual marks	$\mathbf{45.38}_{+9.52}$	$\mathbf{45.21}_{+9.35}$	$\mathbf{19.46}_{+3.48}$

Table 1: **VLMs use feedback to improve grounding accuracy.** We explore how much noise-free Class Label Feedback and Binary Feedback improve semantic grounding in VLMs. Furthermore, we seek the best textual and visual prompt to encourage VLMs to attend and follow the feedback. For each type of feedback and each VLM, we highlight the largest improvements w.r.t. the performance of its base predictions.

The COCO validation set is composed of 5k images. Consistent with prior works on VLM grounding (Yang et al., 2023a), we use the same subset of 100 images for ADE20k and COCO for our analysis. We include the full dataset details in the Appendix B.

VLMs. We analyze three state-of-the-art open-source VLMs including LLaVA-1.5 (Liu et al., 2023a), ViP-LLaVA (Cai et al., 2024) and CogVLM (Wang et al., 2024). LLaVA-1.5 is a successor of LLaVA (Liu et al., 2023b), a visual instruction tuned VLM, and has scaled up to a larger model and a larger training dataset. ViP-LLaVA shares the overall model architecture and training strategy with LLaVA, but focuses on synthesizing a diverse set of visual marks in the training dataset, effectively improving the model performance when using visual prompts and allowing for a more user-friendly interface. CogVLM is a generalist VLM with highlights on integrating image and text features without sacrificing any performance on NLP tasks.[2]

Metrics. We evaluate semantic grounding performance by measuring classification accuracy. We use off-the-shelf text embeddings (Huggingface) to map the VLM outputs o_i to the nearest label from the class label list. We then report accuracy $Acc_{\text{grounding}}$ aggregated over all regions $\mathrm{r_i}$ for each scene in the dataset. We evaluate the ability of a VLM verifier to generate a binary feedback signal by measuring the F_1-score of its binary predictions on whether a set of semantic grounding predictions is accurate.

4 Empirical Findings

In this section, we discuss the results obtained with respect to our research questions after experimenting on the ADE20k. All experiments are run on three different seeds and we report the average performances.

2. We adopt LLaVA-1.5 13b (from `https://huggingface.co/llava-hf/llava-1.5-13b-hf`), ViP-LLaVA 13b (from `https://huggingface.co/llava-hf/vip-llava-13b-hf`), and CogVLM (from `https://huggingface.co/THUDM/CogVLM`)

4.1 Can VLMs Receive Grounding Feedback?

Table 1 summarizes the base predictions for each model and improved grounding accuracy after receiving grounding feedback. We include the full prompt template of each feedback prompt structure in the Appendix A.

Q1 What type of feedback yields the best improvement in grounding accuracy?

We first compare the improvement in accuracy without additional prompting techniques (*i.e.*, without zero-shot CoT or visual prompts in Fig. 3). Table 1 shows that noise-free feedback improve grounding accuracy by up to 61.06 and 5.18 points by taking class label feedback and binary feedback, respectively. We conclude that VLMs can 'read' feedback to improve performance, without requiring any additional data, training time, or architectural modifications.

Intuitively, class label feedback yields the most improvement, since it directly reveals the class labels and consequently reduces the semantic grounding task to a text retrieval problem. Perhaps surprisingly, class label feedback does not automatically improve accuracy to 100%. This outcome points to a limitation in the ability of open-source VLMs to handle tasks relying purely on language understanding, suggesting an area for potential improvement in these models (Lin et al., 2023).

In the rest of this paper, we focus on binary feedback, which is the more practical setting for automated feedback generation.

Q2 How should the binary feedback be prompted to a VLM?

Table 1 shows that zero-shot CoT can augment binary feedback for every model considered by up to 2.26 accuracy points. This observation aligns with trends in LLMs that suggest the effectiveness of CoT to improve reasoning (Wei et al., 2022; Kojima et al., 2022). On the other hand, visual prompting using SoM (Yang et al., 2023a) does not yield significant additional improvement beyond zero-shot CoT for models that were not already pre-trained with data featuring visual prompting cues (*e.g.*, LLaVa-1.5). In contrast, ViP-LLaVA was specifically trained for interpreting visual cues and consequently, improves with both SoM and visual marks (*e.g.*, red circles). Notably, the combination of zero-shot CoT and visual marks emerges as the most effective strategy, yielding a substantial increase of 7.45 grounding accuracy points relative to the base predictions.

We conclude that for open-source VLMs, the best way to introduce binary feedback in semantic grounding is to combine visual marks and zero-shot CoT. Compared to the prior work (Huang et al., 2023b) that explored to use multimodal CoT to improve visual question answering in VLMs, taking binary feedback in semantic grounding requires additional region-level knowledge. When used on a model that has been trained to recognize visual prompting, we can improve grounding accuracy by up to 9.35 points via a single round of feedback. This observation confirms findings from Shtedritski et al. (2023), which suggested that internet-scale trained vision encoders have an inherent tendency to focus on such visual cues.

Approach	Visual prompt	LLaVA-1.5	ViP-LLaVA	CogVLM
Intrinsic Self-Correction	N/A	51.12	48.19	21.87
VLM Binary Verification	Visual marks	56.16	**60.47**	39.16
	RoI crop	**61.71**	58.18	**40.68**
	Visual marks + RoI crop	61.14	59.6	39.79

Table 2: **VLM binary verification provide higher-quality binary feedback compared to intrinsic self-correction.** The choices of visual prompting techniques should be tailored to the specific VLMs. We bold the best performances of each VLM.

4.2 Can VLMs Give Binary Grounding Feedback for Themselves?

Table 2 provides F_1 scores of the binary feedback produced by a VLM Verifier. We include prompt template to produce binary feedback in the Appendix A.

Q3 Can a VLM generate binary feedback via intrinsic self-correction?

We first assess the effectiveness of intrinsic self-correction, which involves continuing another round of conversation by asking *'Carefully review and refine your answer'* to the VLM and directly outputting the revised predictions. We derive the binary feedback by comparing whether the revised predictions differ from the initial predictions. Table 2 shows that the overall F_1 score, evaluated on how often the verifier identifies grounding errors, is low with intrinsic self-correction. When evaluated in accuracy, intrinsic self-correction achieves low accuracies at 47.03, 47.13, and 59.5 on LLaVA-1.5, ViP-LLaVA, and CogVLM, respectively. In Appendix D.1, we further demonstrate that F_1 is a more representative metric for enhancing VLMs with self-generated feedback. This result is aligned with previous studies on LLMs (Huang et al., 2023a) that LLMs struggle to improve via intrinsic self-correction.

Q4 Can a VLM leverage external techniques to bypass the pitfalls of intrinsic self-correction?

External techniques have been shown to improve intrinsic self-correction in LLMs (Chen et al., 2024; Gou et al., 2024). Consequently in Table 2, we propose a binary verification mechanism with VLM using Region-of-Interest (RoI) crop, where the Verifier receives only an RoI crop of an object and must predict whether the proposed label accurately describes the object. Simplifying the intrinsic self-correction problem to a binary classification problem drastically improves F_1 score for all three models, by up to 18.81 points. This observation is well-aligned with the strong self-evaluation capabilities in LLMs (Kadavath et al., 2022). We may also augment this binary verification with visual marks such as red circles. Furthermore, the choice of the visual prompting technique should be tailored to the specific VLM. For instance, crop tends to be more effective for networks not trained on visual marks (*e.g.*, LLaVA-1.5 and CogVLM), while visual marks yield better results for models accustomed to such cues (*e.g.*, ViP-LLaVA).

VLM	Binary feedback source	Dialogue round					
		t = 0	t = 1	t = 2	t = 3	t = 4	t = 5
LLaVA-1.5	Intrinsic Self-Correction	35.86	30.92	29.64	$28.54_{-7.32}$	-	-
	VLM Binary Verification **(ours)**	35.86	37.97	38.93	39.27	39.54	$40.29_{+4.43}$
	Noise-Free Verification **(ours)**	35.86	45.42	47.95	51.55	52.04	$53.2_{+17.34}$
ViP-LLaVA	Intrinsic Self-Correction	35.86	27.72	26.7	$25.68_{-10.18}$	-	-
	VLM Binary Verification **(ours)**	35.86	35.14	36.06	36.37	36.16	$36.47_{+0.39}$
	Noise-Free Verification **(ours)**	35.86	47.45	47.64	50.54	51.82	$53.13_{+17.27}$
CogVLM	Intrinsic Self-Correction	15.98	8.33	8.6	$9.08_{-6.9}$	-	-
	VLM Binary Verification **(ours)**	15.98	17.13	17.96	18.09	18.5	$18.64_{+2.66}$
	Noise-Free Verification **(ours)**	15.98	19.6	20.96	21.51	21.82	$22.12_{+6.14}$
GPT-4V & SoM	Intrinsic Self-Correction	40.36	22.33	25.2	$22.95_{-17.41}$	-	-
	VLM Binary Verification **(ours)**	40.36	41.8	43.23	$42.4_{+2.04}$	-	-
	Noise-Free Verification **(ours)**	40.36	50	52.45	$53.27_{+12.91}$	-	-

Table 3: **Iterative automated feedback improves semantic grounding in ADE20k.** We highlight the performance difference w.r.t. the performance of the base predictions and if the performances are below the performance of the base predictions, we use red-colored font.

5 Towards Iterative Self-Generated Feedback for Semantic Grounding in VLMs

We have established that VLMs can improve on semantic grounding when given (noise-free) feedback. Furthermore, VLMs can also act as Verifiers that produce poentially noisy, binary feedback. We now combine these findings to develop an automated feedback-based semantic grounding framework over multiple rounds of iterative dialogue. Our goal is to highlight the potentials and limitations of a prompt-only framework as a promising approach to improve the semantic understanding of open-source Internet-Scale VLMs, revealing meaningful improvements over initial predictions without any fine-tuning or model engineering.

5.1 Binary *Self-Feedback* in VLMs

We introduce an iterative loop of dialogue between a VLM agent and Verifier, where at the first timestep $\mathrm{t} = 0$, we run the VLM to obtain base predictions $\{\mathrm{o}_{\mathrm{i},0}\}_{\mathrm{i}=1}^{\mathrm{N}}$ for every scene (Sec. 3.1). We then prompt the same VLM to generate a binary feedback signal for every prediction $f^{\mathrm{VLM}}(\mathrm{x}, \mathrm{r}_\mathrm{i}, \mathrm{o}_{\mathrm{i},0})$ (Sec. 3.2.2). In the next time step, we provide this binary feedback to the VLM agent and ask it to re-generate predictions (Sec. 3.2.1). We repeat these steps for a finite number of iterations or until the verifier agrees with the prediction. Fig. 4 demonstrates the iterative interactions between a VLM agent and the Verifier.

In our experiments, we use the textual prompts (*i.e.*, zero-shot CoT) and the visual prompts (*i.e.*, red circles) to encourage feedback receiving and use RoI crop when VLMs provide binary feedback. Consistent with the prior work on VLM grounding (Yang et al., 2023a), we use the same subset of 100 images for ADE20k and COCO for our analysis.

VLM	Binary feedback source	Dialogue round					
		t = 0	t = 1	t = 2	t = 3	t = 4	t = 5
LLaVA-1.5	Intrinsic Self-Correction	36.3	33.69	32.26	$31.63_{-4.66}$	-	-
	VLM Binary Verification **(ours)**	36.3	35.87	36.94	37.04	37.69	$38.21_{+1.91}$
	Noise-Free Verification **(ours)**	36.3	41.55	43.81	46.22	47.55	$48.77_{+12.47}$
ViP-LLaVA	Intrinsic Self-Correction	37.26	32.64	32.4	$31.12_{-6.13}$	-	-
	VLM Binary Verification **(ours)**	37.26	37.84	39.64	39.64	40.01	$40.44_{+3.18}$
	Noise-Free Verification **(ours)**	37.26	44.9	48.08	50.15	51.75	$52.54_{+15.28}$
CogVLM	Intrinsic Self-Correction	14.8	16.23	16.47	$15.92_{+1.11}$	-	-
	VLM Binary Verification **(ours)**	14.8	16.97	17.83	18.3	18.52	$18.84_{+4.04}$
	Noise-Free Verification **(ours)**	14.8	19.42	20.14	20.7	21.01	$21.25_{+6.45}$
GPT-4V & SoM	Intrinsic Self-Correction	40.92	30.89	36.62	$32.8_{-8.12}$	-	-
	VLM Binary Verification **(ours)**	40.92	43.94	44.9	$45.38_{+4.46}$	-	-
	Noise-Free Verification **(ours)**	40.92	52.7	56.5	$57.8_{+16.88}$	-	-

Table 4: **Iterative automated feedback improves semantic grounding in COCO.** We highlight the performance difference w.r.t. the performance of the base predictions and if the performances are below the performance of the base predictions, we use red-colored font.

Baselines. We compare our approach, referred to as VLM binary verification with an alternative feedback-based approach that is inspired by prior work in LLMs (Kim et al., 2023; Huang et al., 2023a) and uses only intrinsic self-correction. We also compare to a Noise-Free approach, where the feedback is provided by an oracle and assumed to be perfect; the latter acts as an upper bound achievable only with access to an oracle.

Models. We perform the experiments on all three open-source models previously studied. We further experiment our framework using GPT-4V, a state-of-the-art closed-source model. As suggested in prior work (Yang et al., 2023a,c), GPT-4V tends to exhibit better grounding ability when the objects are specified by visual prompts rather than text prompts. Therefore, we adopt GPT-4V & SoM to obtain base predictions, where we overlay object masks and numeric identifiers on the images. In addition, when using VLMs to produce feedback, we apply SoM to specify each object. Finally, since GPT-4V has a longer context window compared to open-source VLMs, we include the class list in the prompt to encourage better alignment between the responses and the ground truth. All GPT-4V experiments are done over the OpenAI API and we follow the exact same evaluation procedures described in Sec. 3.3, where we use the off-the-shelf text embeddings (Huggingface) to map the GPT-4V outputs $\mathrm{o_i}$ to the nearest label from the class label list. We provide further implementation details in Appendix C.

5.2 Discussion

Table 3 and Table 4 show that noise-free feedback provided over multiple rounds can consistently improve all open-source VLMs, with gains ranging from 6.14 to 17.34 for ADE20k and 6.45 to 15.28 in COCO, respectively. Furthermore, multiple feedback rounds improves

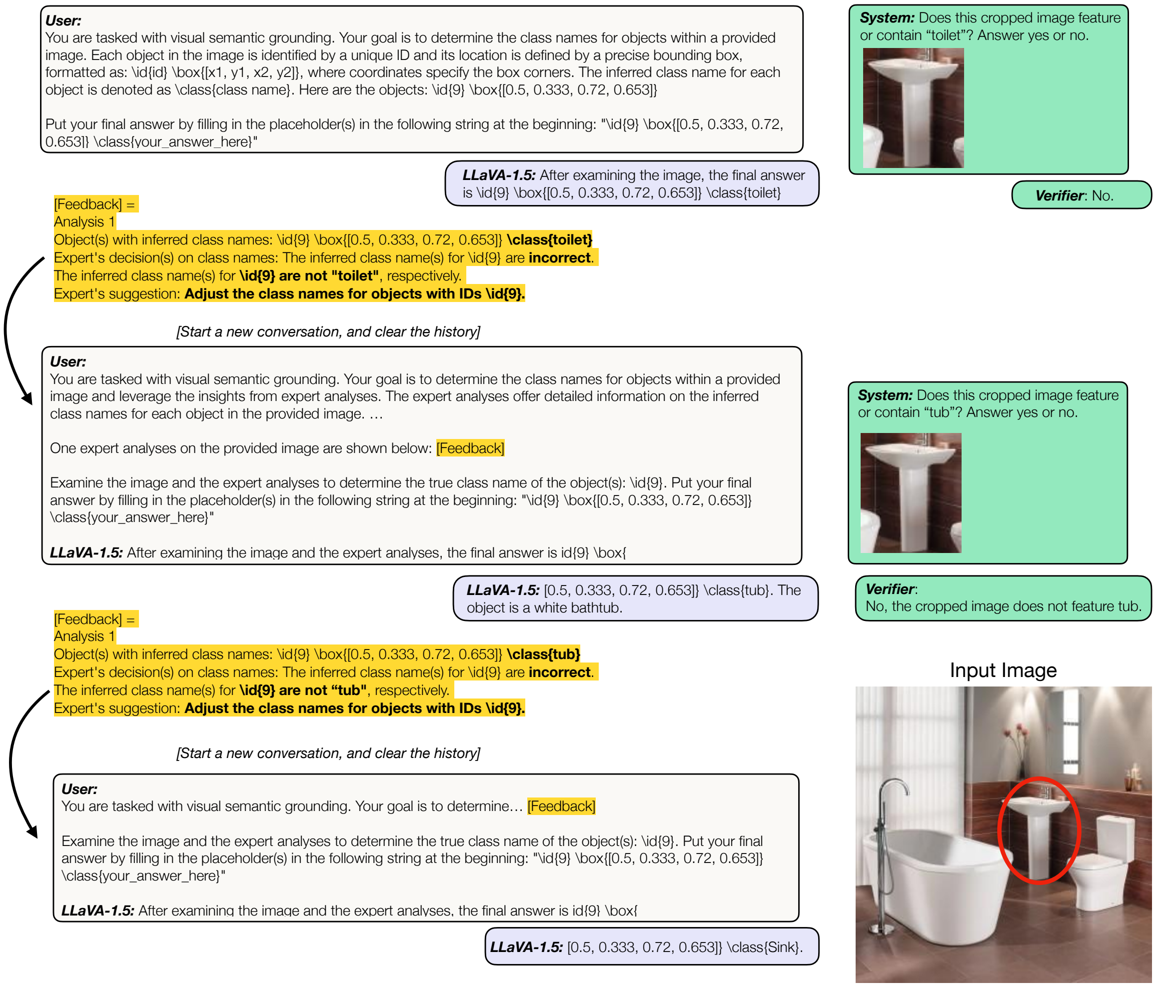

Figure 4: **Example dialogue of using iterative binary self-feedback to improve semantic grounding in VLMs.** Initially, the user queries the semantic class of an object within a bounding box. At the first attempt, the VLM responds without feedback. To refine accuracy, we use the verifier (the same VLM) to answer a yes/no question for binary self-feedback. Incorporating this feedback, we prompt the VLM again, leading to a refined prediction. The VLM's initial guess evolves from 'toilet' to 'bathtub', and ultimately to 'sink' – the correct classification.

grounding accuracy by up to 7.78 and 7.64 additional points on ADE20k and COCO, respectively, when compared to a single feedback round (*i.e.*, t = 1). Finally, our binary verification mechanism, although yielding noisy feedback, also consistently improves grounding accuracy between 0.39 to 4.43 points for ADE20k and between 1.91 and 4.04 points for COCO. We emphasize that these gains are consistent across all models, demonstrating the value of iterative feedback to yield zero-shot improvements in grounding accuracy, *even when the feedback is noisy.*

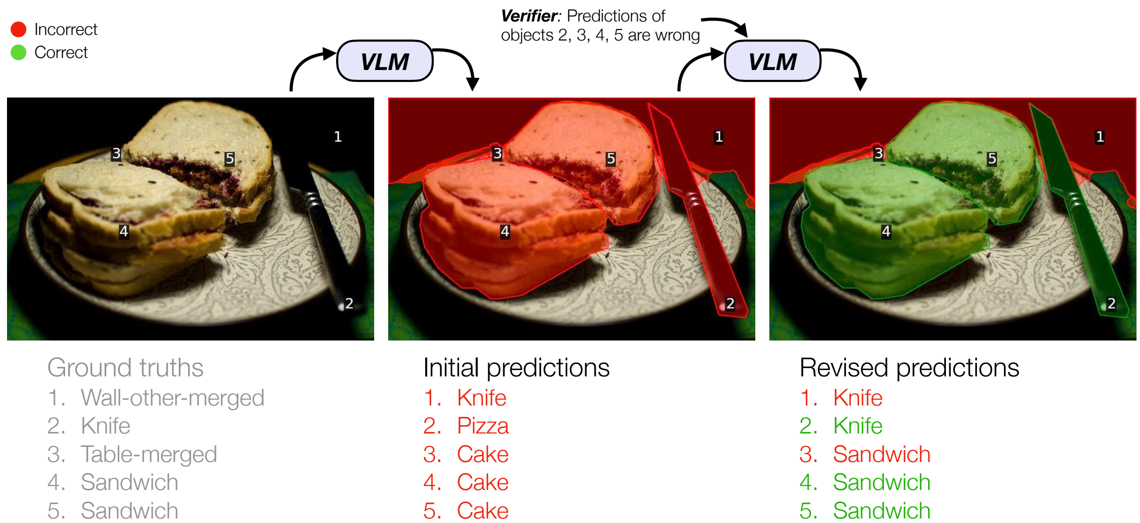

Figure 5: **Enhancing semantic grounding in VLMs with self-generated feedback.** We use GPT-4V & SoM as the VLM here. From the left to the center figure, GPT-4V takes the SoM-prompted image Yang et al. (2023a) as input and struggles to predicts the class names of each object. From the center to the right figure, GPT-4V takes the same SoM-prompted image and the additional feedback from the verifier and successfully correct the class names of three out of five objects. The verifier is another GPT-4V that operates on an altered input image and may produce noisy feedback, *e.g.*, misclassify object 1 as correct.

Intrinsic self-correction decreases downstream grounding accuracy by up to 10.18 in ADE20k. In comparison, our feedback mechanism leverages two key tools: (i) modifying the input image via an RoI crop and markings, (ii) a binary self-generated feedback signal on whether the verifier believes that a given label is correct. Combining these two steps effectively mitigates the self-bias in VLMs that prevents intrinsic self-correction.

For GPT-4V & SoM [3], the results are consistent with open-source VLMs. We first observe that GPT-4V & SoM leads to the better base predictions, *i.e.*, t = 0. Furthermore, GPT-4V & SoM improves 4.46 and 16.88 in COCO when taking VLM binary self-feedback and noise-free feedback, respectively. This suggests that GPT-4V is a better model in receiving binary feedback, potentially due to its larger context window, as compared to open-source VLMs. Fig. 5 shows the effectiveness of VLM binary verification in GPT-4V. [4]

We highlight that the introduced binary self-feedback requires no additional training, architectural modifications, or in-domain data. We believe these to be very encouraging results showing the potential of this mechanism on VLMs, and opening further research questions on the quest for better automated verification protocols for computer vision tasks. We also anticipate that improvement from iterative feedback will improve with better VLMs.

3. Assessed in Mar 2024 via OpenAI API.
4. GPT-4V & SoM predictions with simplified prompts as of Mar 22, 2024: `https://imgur.com/a/f64eCB3`

6 Conclusion

In this work, we explore the potential and challenges of using prompt-based feedback to enhance semantic grounding in VLMs. By addressing two fundamental questions, (i) Can VLMs effectively incorporate grounding feedback? and (ii) Are VLMs capable of providing feedback?, we systematically analyze the most appropriate way to prompt the VLMs for each task. We study these two questions on three state-of-the-art VLMs: LLaVA-1.5, ViP-LLaVA, and CogVLM. We observe that VLMs notably improve their grounding performance even with a simple binary feedback. Perhaps surprising, we show these VLMs cannot optimally utilize the class label feedback, even if noise-free, suggesting that challenges in open-source VLMs on tasks that rely purely on language retrieval exists. On the other hand, we show that it is possible to overcome the self-bias tendency on VLMs using a simple binary verification mechanism which produces notably higher-quality feedback compared to intrinsic self-correction. Finally, we combine our empirical findings to develop an automated feedback-based semantic grounding framework over multiple rounds of iterative dialogue. Within five rounds of noise-free feedback, our framework improves semantic grounding in ADE20k split and COCO split up to 17 and 15 accuracy points, respectively. When we switch to the automated feedback regime, the feedback from VLM binary verification outperforms intrinsic self-correction by 7.7 and 2.7 on average in ADE20k split and COCO split, respectively. Furthermore, when compared with their base predictions, VLM binary verification improves up to nearly 5 accuracy points. We further validate this framework on GPT-4V with various feedback mechanisms and find that our observations are consistent across both open-source and proprietary VLMs, further highlighting the importance of using feedback to improve semantic grounding.

Limitations. Despite the improvements in VLMs semantic grounding via prompt-based feedback, this setting trades performances with compute. Therefore, in applications requiring low latency, feedback-based reasoning becomes less practical. Additionally, evaluating VLMs' zero-shot capabilities with the close-set vocabularies amplifies the issue of language ambiguities. For example, in ADE20k, there are classes that are extremely similar, *e.g.*, 'grass', 'hill', 'field', 'plant', and 'tree'. For proprietary VLMs, we include the class list to the prompt; however, it does not solve the ambiguity issue since every dataset might have different class interpretations or taxonomies. For open-source VLMs, due to the relatively small context window, we leverage the off-the-shelf embeddings for such mapping, which can be noisy. Finally, we notice the open-source VLMs struggle if tasked at the same time to listen or provide feedback involving multiples regions of the image. We believe this might be related to the datasets used for training mostly containing closely related objects for grounding and recognition questions. We anticipate the future generations of open-source VLMs will yield substantial quantitative advancements in the task at hand.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee. Making large multimodal models understand arbitrary visual prompts. *arXiv preprint arXiv:2312.00784*, 2023.

Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P. Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee. Making large multimodal models understand arbitrary visual prompts. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024.

Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic, 2023.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=KuPixIqPiq`.

Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou, Kaizhao Liang, Jintai Chen, Juanwu Lu, Zichong Yang, Kuei-Da Liao, et al. A survey on multimodal large language models for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 958–979, 2024.

Yin Cui, Guandao Yang, Andreas Veit, Xun Huang, and Serge Belongie. Learning to evaluate image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate, 2024. URL `https://openreview.net/forum?id=QAwaaLJNCk`.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=Sx038qxjek`.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=lL3lnMbR4WU.

Qiushan Guo, Shalini De Mello, Hongxu Yin, Wonmin Byeon, Ka Chun Cheung, Yizhou Yu, Ping Luo, and Sifei Liu. Regiongpt: Towards region understanding vision language model. *arXiv preprint arXiv:2403.02330*, 2024.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIP-Score: A reference-free evaluation metric for image captioning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.595. URL https://aclanthology.org/2021.emnlp-main.595.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet, 2023a.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, Qiang Liu, Kriti Aggarwal, Zewen Chi, Johan Bjorck, Vishrav Chaudhary, Subhojit Som, Xia Song, and Furu Wei. Language is not all you need: Aligning perception with language models, 2023b.

Huggingface. sentence-transformers/all-mpnet-base-v2. https://huggingface.co/sentence-transformers/all-mpnet-base-v2.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zachary Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, John Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom B. Brown, Jack Clark, Nicholas Joseph, Benjamin Mann, Sam McCandlish, Christopher Olah, and Jared Kaplan. Language models (mostly) know what they know. *ArXiv*, abs/2207.05221, 2022. URL https://api.semanticscholar.org/CorpusID:250451161.

Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. Re-evaluating automatic metrics for image captioning. In Mirella Lapata, Phil Blunsom, and Alexander Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 199–209, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-1019.

Geunwoo Kim, Pierre Baldi, and Stephen Marcus McAleer. Language models can solve computer tasks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=M6OmjAZ4CX`.

Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollar. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=e2TBb5y0yFf`.

Byung-Kwan Lee, Beomchan Park, Chae Won Kim, and Yong Man Ro. Collavo: Crayon large language and vision model, 2024.

Bo Li, Peiyuan Zhang, Jingkang Yang, Yuanhan Zhang, Fanyi Pu, and Ziwei Liu. Otterhd: A high-resolution multi-modality model, 2023.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models, 2024.

Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models, 2023.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL `http://arxiv.org/abs/1405.0312`.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023b.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next/`.

Yujie Lu, Xianjun Yang, Xiujun Li, Xin Eric Wang, and William Yang Wang. Llmscore: Unveiling the power of large language models in text-to-image synthesis evaluation, 2023.

Yujie Lu, Xianjun Yang, Xiujun Li, Xin Eric Wang, and William Yang Wang. Llmscore: Unveiling the power of large language models in text-to-image synthesis evaluation. *Advances in Neural Information Processing Systems*, 36, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=S37hOerQLB`.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, Anton Belyi, Haotian Zhang, Karanjeet Singh, Doug Kang, Hongyu Hè, Max Schwarzer, Tom Gunter, Xiang Kong, Aonan Zhang, Jianyu Wang, Chong Wang, Nan Du, Tao Lei, Sam Wiseman, Mark Lee, Zirui Wang, Ruoming Pang, Peter Grasch, Alexander Toshev, and Yinfei Yang. Mm1: Methods, analysis & insights from multimodal llm pre-training, 2024.

Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, Quan Vuong, Tingnan Zhang, Tsang-Wei Edward Lee, Kuang-Huei Lee, Peng Xu, Sean Kirmani, Yuke Zhu, Andy Zeng, Karol Hausman, Nicolas Heess, Chelsea Finn, Sergey Levine, and Brian Ichter. Pivot: Iterative visual prompting elicits actionable knowledge for vlms, 2024a.

Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. *arXiv preprint arXiv:2402.07872*, 2024b.

Deepak Nathani, David Wang, Liangming Pan, and William Yang Wang. MAF: Multi-aspect feedback for improving reasoning in large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL `https://openreview.net/forum?id=bNeDLx5O6w`.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*, 2023.

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

Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. What does clip know about a red circle? visual prompt engineering for vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11987–11997, October 2023.

C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 100(3/4):441–471, 1987. ISSN 00029556. URL http://www.jstor.org/stable/1422689.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, and Radu Soricut. Gemini: A family of highly capable multimodal models, 2023.

Gemini Team, Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, and Timothy Lillicrap et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, and Yasmine Babaei et al. Llama 2: Open foundation and fine-tuned chat models, 2023.

Gladys Tyen, Hassan Mansoor, Victor Cărbune, Peter Chen, and Tony Mak. Llms cannot find reasoning errors, but can correct them!, 2024.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. Cogvlm: Visual expert for pretrained language models, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_VjQlMeSB_J.

Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas Guibas, Dahua Lin, and Gordon Wetzstein. Gpt-4v(ision) is a human-aligned evaluator for text-to-3d generation, 2024.

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Yang Wang. Perils of self-feedback: Self-bias amplifies in large language models, 2024.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v, 2023a.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023b.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v(ision), 2023c.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Deliberate problem solving with large language models, 2023.

Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity, 2023.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions, 2016.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=KRLUvxh8uaX.

Yuchen Zeng, Wonjun Kang, Yicong Chen, Hyung Il Koo, and Kangwook Lee. Can mllms perform text-to-image in-context learning?, 2024.

Michael R Zhang, Nishkrit Desai, Juhan Bae, Jonathan Lorraine, and Jimmy Ba. Using large language models for hyperparameter optimization. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023a.

Shilong Zhang, Peize Sun, Shoufa Chen, Minn Xiao, Wenqi Shao, Wenwei Zhang, Yu Liu, Kai Chen, and Ping Luo. GPT4roi: Instruction tuning large language model on region-of-interest, 2024. URL https://openreview.net/forum?id=DzxaRFVsgC.

Xinlu Zhang, Yujie Lu, Weizhi Wang, An Yan, Jun Yan, Lianke Qin, Heng Wang, Xifeng Yan, William Yang Wang, and Linda Ruth Petzold. Gpt-4v (ision) as a generalist evaluator for vision-language tasks. *arXiv preprint arXiv:2311.01361*, 2023b.

Xinlu Zhang, Yujie Lu, Weizhi Wang, An Yan, Jun Yan, Lianke Qin, Heng Wang, Xifeng Yan, William Yang Wang, and Linda Ruth Petzold. Gpt-4v(ision) as a generalist evaluator for vision-language tasks, 2023c.

Haozhe Zhao, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang. MMICL: Empowering vision-language model with multi-modal in-context learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5KojubHBr8.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. Step-back prompting enables reasoning via abstraction in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3bq3jsvcQ1.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=WZH7099tgfM`.

Appendix A. Prompt Template

We show the full prompt template for 1) producing base semantic grounding predictions in Fig.6, 2) enhancing previous semantic grounding predictions by taking feedback in Fig. 7 and Fig. 8, and 3) producing self-generated feedback in Fig. 9. For the GPT-4V experiments, we provide the class names by appending *You must answer by selecting from the following names: [COCO or ADE20k Vocabulary]* in the prompt[5], as shown in Fig. 10 and Fig. 11.

```
User: You are tasked with visual semantic grounding. Your goal is to determine the class names for
objects within a provided image. Each object in the image is identified by a unique ID and its location is
defined by a precise bounding box, formatted as: \id{id} \box{box_format}, where coordinates specify the
 box corners. The inferred class name for each object is denoted as \class{class name}. Here are the
objects: obj_ids_str
Put your final answer by filling in the placeholder(s) in the following string at the beginning: "
output_format"

% Example below
% box_format = [x1, y1, x2, y2]
% obj_ids_str = \id{2} \bbox{[0.1, 0.2, 0.13, 0.43]}
% output_format = \id{2} \bbox{[0.1, 0.2, 0.13, 0.43]} \class{your_answer_here}
```

Figure 6: Prompt template to produce the base predictions. Text colored in blue denote variables.

Appendix B. Dataset Details

We use ADE20k and COCO panoptic segmentation dataset to evaluate the semantic grounding performance in VLMs. We adopt SoM split provided in the prior work Yang et al. (2023a)[6]. ADE20k is a large-scale dataset with fine-grained segmentation labels. We adopt the variant with 150 classes, commonly referred to as ADE20k-150. COCO panoptic segmentation is a standard dataset to evaluate visual grounding. There are 133 fine-grained classes in total, composed of 80 thing classes and 53 stuff classes. Consistent with prior works, SoM (Yang et al., 2023a), we use the same subset of 100 images for ADE20k and COCO for our analysis. There are 100 images and 488 segmentation masks in ADE20k SoM split and 101 and 628 segmentation masks in COCO SoM split.

Appendix C. Implementation Details

Every experiment throughout this paper is run over three seeds and we report the average scores except for experiments with proprietary VLMs. In the experiments with binary or class label feedback, we only ask VLM to correct those that are incorrect based on the feedback. Therefore, if the feedback is noisy, *e.g.* VLM binary verification, VLM can possibly decrease the performances. See Fig. 15 for example.

5. `https://github.com/microsoft/SoM/tree/main/benchmark#open-vocab-segmentation-on-coco`
6. `https://github.com/microsoft/SoM/tree/main/benchmark#dataset`

```
User: You are tasked with visual semantic grounding. Your goal is to determine the class names for
objects within a provided image and leverage the insights from expert analyses. The expert analyses offer
detailed information on the inferred class names for each object in the provided image. Each object in the
image is identified by a unique ID and its location is defined by a precise bounding box, formatted as: \id
{id} \box{box_format}, where coordinates specify the box corners. The inferred class name for each object
 is denoted as \class{class name}. I have labeled each object with its ID and overlaid its segmentation
mask on the image to clarify the correspondences.

One expert analyses on the provided image are shown below:
∗ Analysis 1
Object(s) with inferred class names: previous_predictions
Expert's decision(s) on class names: The inferred class name(s) for incorrect_obj_id are incorrect. The
inferred class name(s) for incorrect_obj_id are not "incorrect_obj_class_name", respectively.
Expert's suggestion: Adjust the class names for objects with IDs incorrect_obj_id.

Examine the image and the expert analyses to determine the true class name of the object(s):
incorrect_obj_id. Put your final answer by filling in the placeholder(s) in the following string at the
beginning: "output_format"

% Example below
% box_format = [x1, y1, x2, y2]
% previous_predictions = \id{6} \box{[0.53, 0.291, 0.998, 0.483]} \class{wall}
% incorrect_obj_id = \id{6}
% incorrect_obj_class_name = wall
% output_format = \id{6} \box{[0.53, 0.291, 0.998, 0.483]} \class{your_answer_here}
```

Figure 7: Prompt template to improve semantic grounding predictions by taking Binary Feedback. Text colored in blue denote variables.

For open-source VLMs, when perform the VLM forward pass $o_i = \mathrm{VLM}(x, r_i, q)$, we set the temperature to 0.9, top_p to 0.8, max_new_tokens to 1024, and draw five samples per forward pass. We take the majority vote responses as the final answers o_i.

For GPT-4V, we follow the inference code provided in Yang et al. (2023a)[7] and set the system prompt as: - *For any marks mentioned in your answer, please highlight them with [].* Every region r_i in ADE20k and COCO panoptic segmentation dataset is represented with segmentation mask. We convert them to a more compact representation, *i.e.* bounding box, and feed them to the VLMs in the text prompt. We follow Yang et al. (2023a) to set the alpha parameters in SoM as 0.2 and 0.4 in ADE20k and COCO, respectively.

Appendix D. Additional Results

D.1 Feedback Accuracy does not Strongly Correlate with Semantic Grounding with Iteratively Self-Generated Feedback

In the main paper, we measure feedback in F_1 score. Another intuitive evaluation metric is feedback accuracy $Acc_{feedback}$. However, we find that VLM binary verification with a

7. `https://github.com/microsoft/SoM/blob/main/gpt4v.py`

User: You are tasked with visual semantic grounding. Your goal is to determine the class names for objects within a provided image and leverage the insights from expert analyses. The expert analyses offer detailed information on the inferred class names for each object in the provided image. Each object in the image is identified by a unique ID and its location is defined by a precise bounding box, formatted as: \id {id} \box{box_format}, where coordinates specify the box corners. The inferred class name for each object is denoted as \class{class name}. I have labeled each object with its ID and overlaid its segmentation mask on the image to clarify the correspondences.

One expert analyses on the provided image are shown below:
∗ Analysis 1
Object(s) with inferred class names: previous_predictions
Expert's decision(s) on class names: The inferred class name(s) for incorrect_obj_id are incorrect. The inferred class name(s) for incorrect_obj_id are not "incorrect_obj_class_name", respectively.
Expert's suggestion: Adjust the class names for objects with IDs incorrect_obj_id to incorrect_obj_ground_truth_class_name, respectively.

Examine the image and the expert analyses to determine the true class name of the object(s): incorrect_obj_id. Put your final answer by filling in the placeholder(s) in the following string at the beginning: "output_format"

% Example below
% box_format = [x1, y1, x2, y2]
% previous_predictions = \id{6} \box{[0.53, 0.291, 0.998, 0.483]} \class{wall}
% incorrect_obj_id = \id{6}
% incorrect_obj_class_name = wall
% incorrect_obj_ground_truth_class_name = house
% output_format = \id{6} \box{[0.53, 0.291, 0.998, 0.483]} \class{your_answer_here}

Figure 8: Prompt template to improve semantic grounding predictions by taking Class Label Feedback. Text colored in blue denote variables.

User: Does this visual_prompt contain "class_name"? Answer yes or no.

% Example below
% visual_prompt = cropped image
% class_name = window

Figure 9: Prompt template to derive VLM binary feedback. Text colored in blue denote variables.

higher $Acc_{feedback}$ does not necessary lead to a higher $Acc_{grounding}$ in the iterative setup in Sec. 5. On average, we find that $Acc_{feedback}$ achieve an 0.11 Spearman rank correlation coefficient Spearman (1987) with $Acc_{grounding}$ at t = 3 as compared to 0.72 achieved by F_1. We conclude that F_1 is a better evaluation metric for measure feedback quality in this work.

User: I have labeled a bright numeric ID at the center for each visual object in the image. Please enumerate their names. You must answer by selecting from the following names:[ADE_class_list]

Figure 10: Prompt template for GPT-4V to produce the base predictions. Following prior work (Yang et al., 2023a), we include the full class list in the text prompt. Text colored in blue denote variables.

User: You are tasked with visual semantic grounding. Your goal is to determine the class names for objects within a provided image and leverage the insights from expert analyses. The expert analyses offer detailed information on the inferred class names for each object in the provided image. Each object in the image is identified by a unique ID and its location is defined by a precise bounding box, formatted as: \id {id} \box{box_format}, where coordinates specify the box corners. The inferred class name for each object is denoted as \class{class name}. I have labeled each object with its ID and overlaid its segmentation mask on the image to clarify the correspondences.

One expert analyses on the provided image are shown below:
∗ Analysis 1
Object(s) with inferred class names: previous_predictions
Expert's decision(s) on class names: The inferred class name(s) for incorrect_obj_id are incorrect. The inferred class name(s) for incorrect_obj_id are not "incorrect_obj_class_name", respectively.
Expert's suggestion: Adjust the class names for objects with IDs incorrect_obj_id.

Examine the image and the expert analyses to determine the true class name of the object(s): incorrect_obj_id. Put your final answer by filling in the placeholder(s) in the following string at the beginning: "output_format" You must answer by selecting from the following names: [ADE_class_list]

% Example below
% box_format = [x1, y1, x2, y2]
% previous_predictions = \id{6} \box{[0.53, 0.291, 0.998, 0.483]} \class{wall}
% incorrect_obj_id = \id{6}
% incorrect_obj_class_name = wall
% output_format = \id{6} \box{[0.53, 0.291, 0.998, 0.483]} \class{your_answer_here}

Figure 11: Prompt template for GPT-4V to improve semantic grounding predictions by taking Binary Feedback. Following prior work Yang et al. (2023a), we include the full class list in the text prompt. Text colored in blue denote variables.

D.2 Qualitative Results

We share additional qualitative results on ADE20k and COCO in Fig. 12, Fig. 13, Fig. 14. We also note that most of the failure cases occur when 1) the VLMs keep their own predictions even though the feedback refers them as incorrect predictions or 2) when the self-generated feedback is incorrect, as shown in Fig. 15.

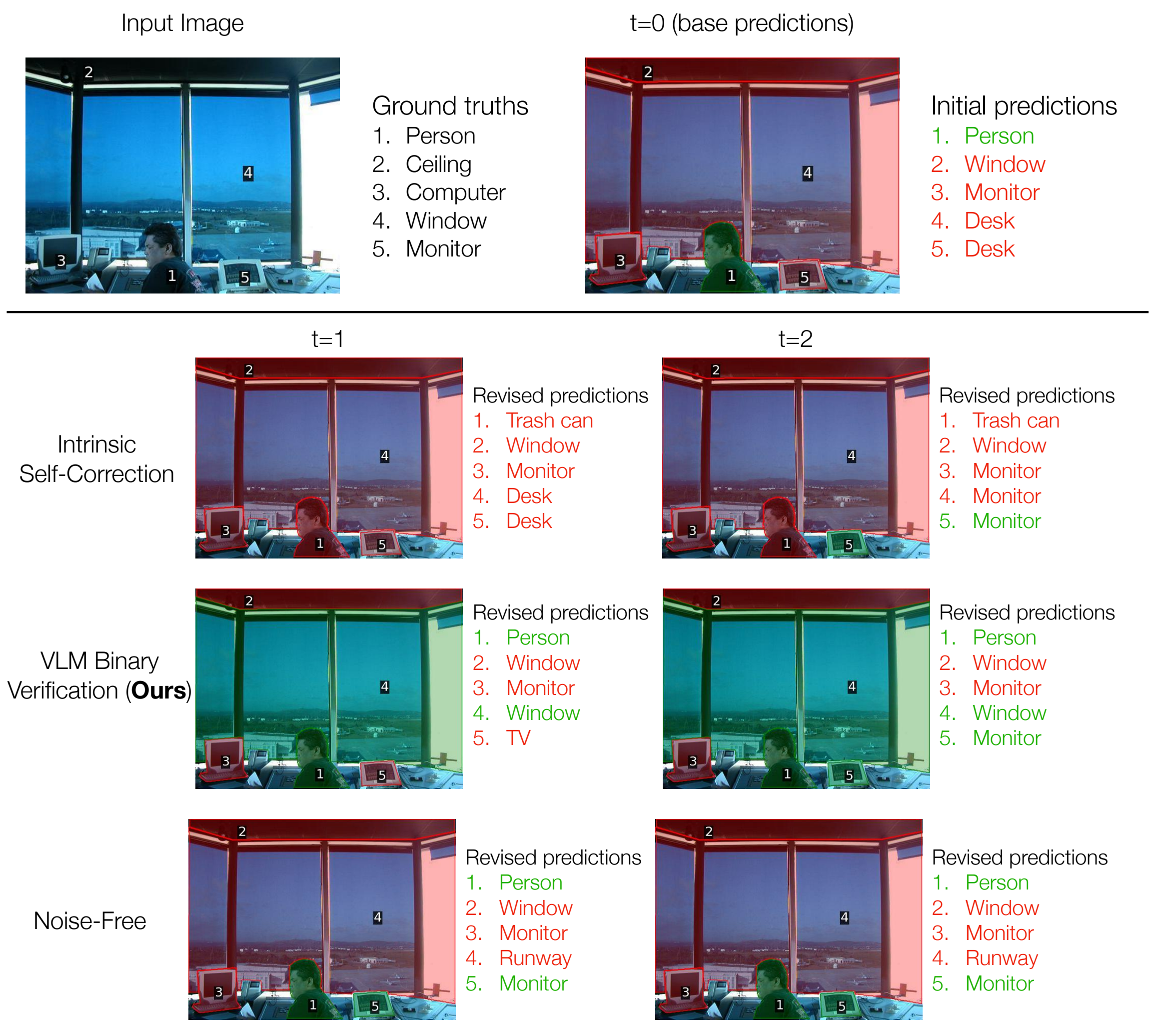

Figure 12: **LLaVA-1.5 qualitative results on ADE20k.** We visualize the predictions of LLaVA-1.5 at time steps from 0 to 2. Intrinsic self-correction fails to identify which predictions are correct/incorrect, while VLM binary verification and Noise-free feedback provide explicit signal on each region, leading to a better chance of correction. From t = 0 to t = 1, we find that VLM might produce different results (object 4) even when receiving the same feedback (VLM binary verification and Noise-free). As explained in Appendix C, in the VLMs forward pass, we draw multiple sequences and take the majority vote as the final responses. For the sake of visualization, we put a bright ID on each object and highlight the incorrect predictions in red and the correct predictions in green.

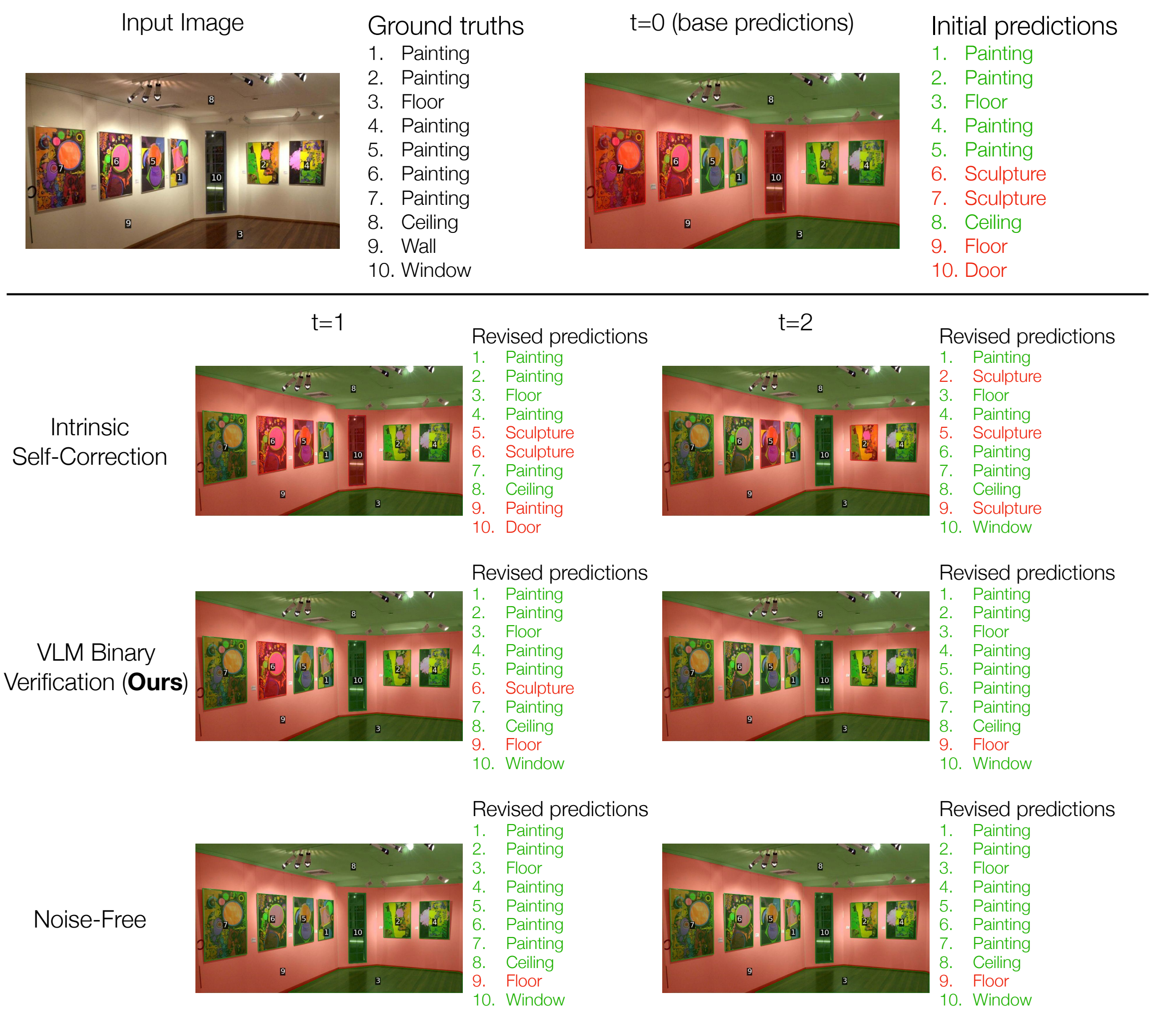

Figure 13: **ViP-LLaVA qualitative results on ADE20k.** We visualize the predictions of ViP-LLaVA at time steps from 0 to 2. Intrinsic self-correction fails to identify which predictions are correct/incorrect, while VLM binary verification and Noise-free feedback provide explicit signal on each region, leading to a better chance of correction. Note that we draw multiple samples in the VLM forward pass, therefore, leading to slightly different results even when the image and query are the same (See Appendix C). For the sake of visualization, we put a bright ID on each object and highlight the incorrect predictions in red and the correct predictions in green.

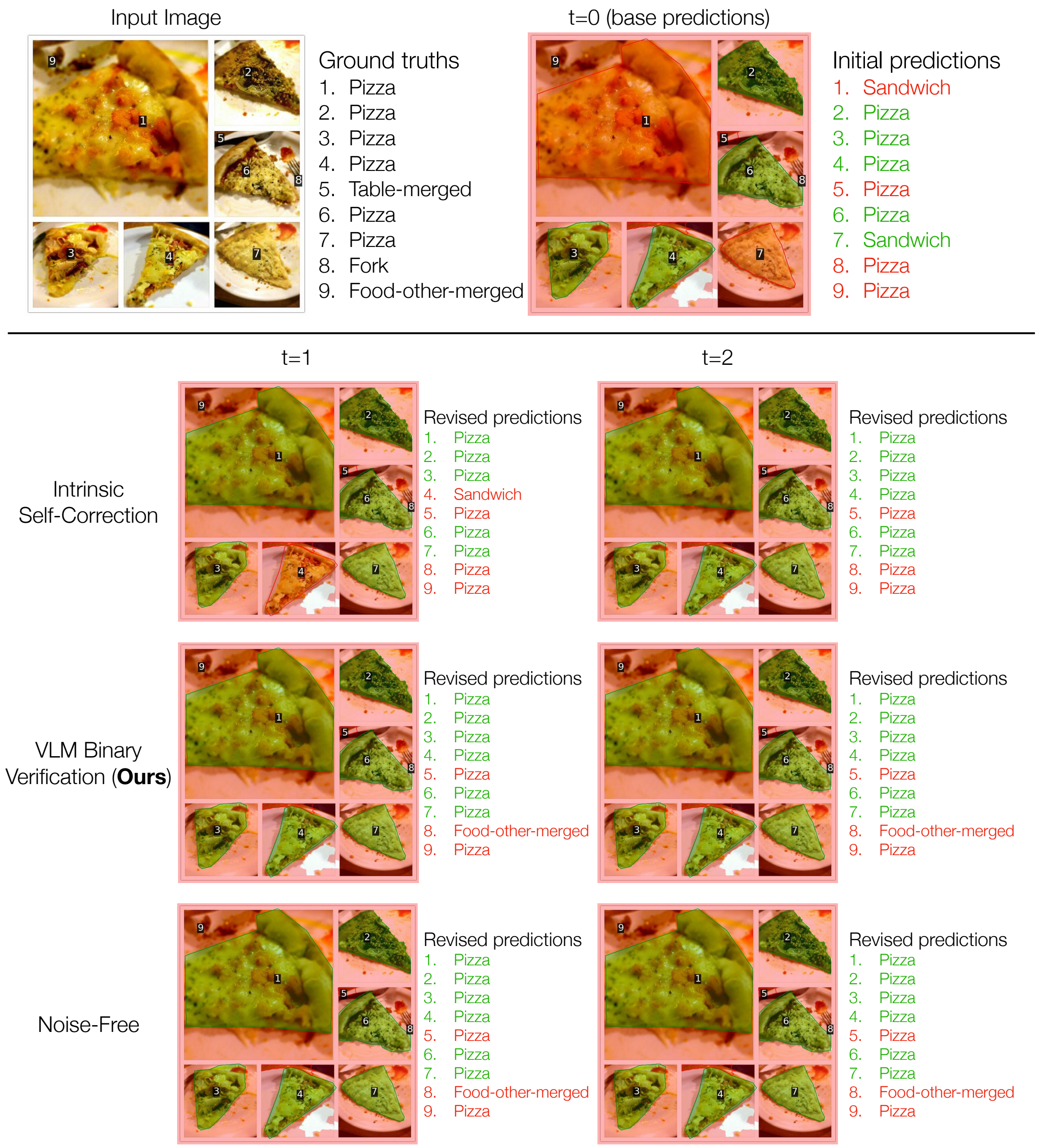

Figure 14: **CogVLM qualitative results on COCO.** We visualize the predictions of CogVLM at time steps from 0 to 2. For the sake of visualization, we put a bright ID on each object and highlight the incorrect predictions in red and the correct predictions in green.

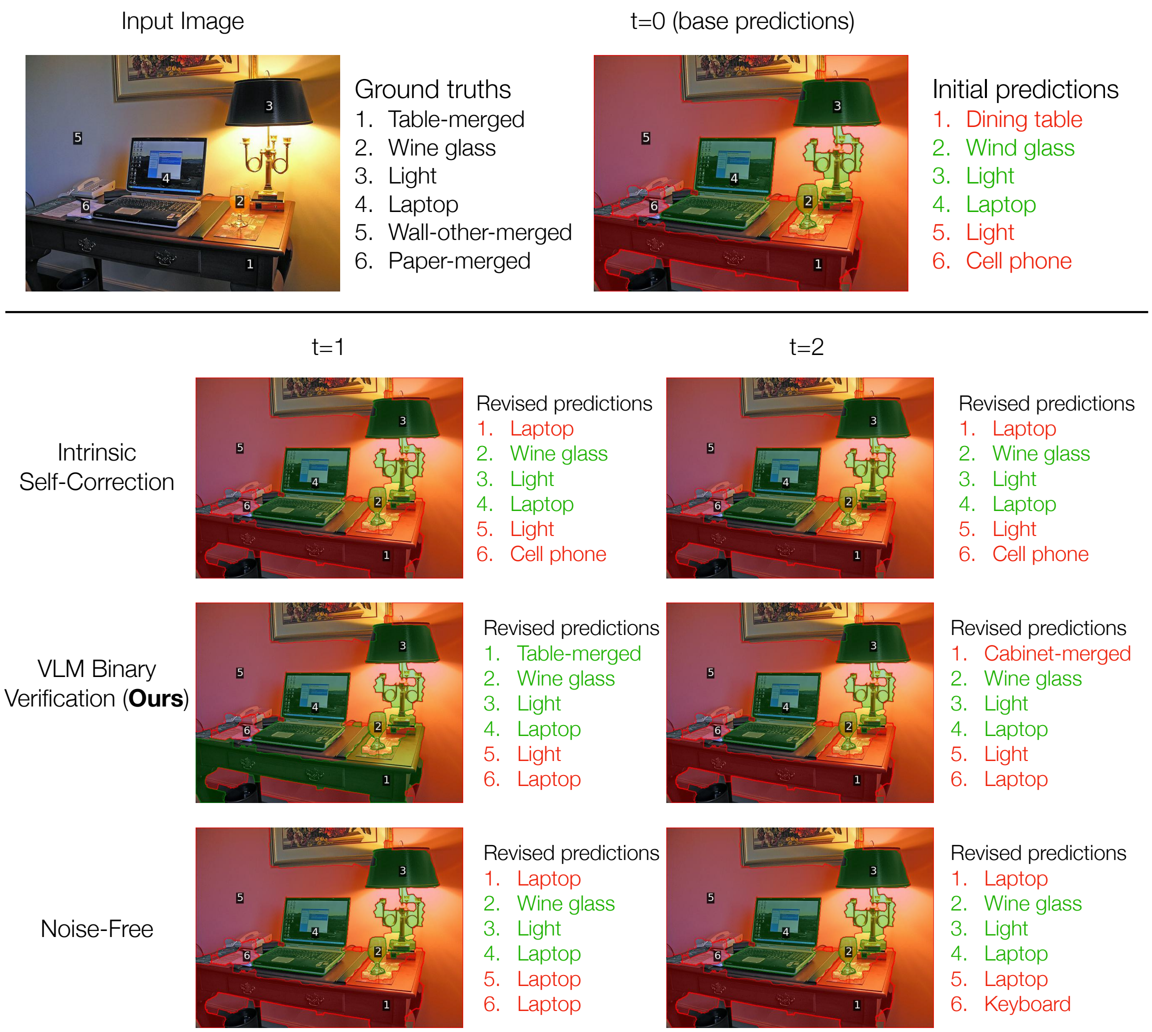

Figure 15: **[Failure case study] LLaVA-1.5 qualitative results on COCO.** All three approaches cannot fix the errors in the initial predictions. For VLM binary verification, from t = 1 to t = 2, the predictions changes from correct (table-merged) to incorrect (cabinet-merged) since the VLM verifier is not perfect and, therefore, providing misleading feedback. Even with the noise-free feedback, LLaVA-1.5 struggle to adjust the predictions. For the sake of visualization, we put a bright ID on each object and highlight the incorrect predictions in red and the correct predictions in green.