SELF-IMPROVING VLM JUDGES WITHOUT HUMAN ANNOTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Effective judges of Vision-Language Models (VLMs) are crucial for model development. Current methods for training VLM judges mainly rely on large-scale human preference annotations. However, such an approach is costly, and the annotations easily become obsolete as models rapidly improve. In this work, we present a framework to self-train a VLM judge model without any human preference annotations, using only self-synthesized data. Our method is iterative and has three stages: (1) generate diverse multimodal instruction-response pairs at varying quality levels, (2) generate reasoning traces and judgments for each pair, removing the ones that do not match our expected quality levels, and (3) training on correct judge answers and their reasoning traces. We evaluate the resulting judge on Multimodal Reward-Bench and VL-RewardBench across domains: correctness, preference, reasoning, safety, and visual question-answering. Our method improves a Llama-3.2-11B multimodal judge from 0.38 to 0.51 in overall accuracy on VL-RewardBench, often outperforming much larger models including Llama-3.2-90B, GPT-4o, and Claude 3.5 Sonnet, with strong gains in general, hallucination, and reasoning dimensions. The strength of these human-annotation-free results suggest the potential for a future self-judge that evolves alongside rapidly improving VLM capabilities.

1 INTRODUCTION

VLM reward models are critical for evaluating output quality and enabling alignment with human preferences Yu et al. (2025); Jing & Du (2024); Sun et al. (2023a); Chen et al. (2024). Existing approaches for training such model evaluators have primarily followed two directions: scaling up human preference collection and distilling from large closed-source models such as GPT and Claude (which also indirectly rely on significant human annotation) Li et al. (2024c); Xiong et al. (2025); Zhang et al. (2023); Zhou et al. (2024a). However, extensive human preference annotation is both costly and labor-intensive, and becomes obsolete as models advance and new tasks arise.

In this paper, we show that it is possible to self train a VLM judge given only the VLM's own generations. Our method requires no human preference annotations, thereby significantly reducing the cost of reward model training. Our key idea is to use simple, general purpose heuristics to construct VLM outputs with varied quality levels, which provide enough signal to train a judge. Our approach follows a three-step iterative process:

- **Synthetic preference pair generation.** We create synthetic preference pairs tailored to different question types. For questions with closed-ended answers (e.g. multiple choice) we generate many candidate responses and pair the majority answer with a random alternative. For open-ended questions (e.g. captioning) we generate responses and deliberately inject meaningful errors into one version, such as changing object attributes or spatial relationships. We create substantial differences rather than minor edits, exposing the model to realistic evaluation scenarios.
- **Judge training data generation.** We use the previous-iteration judge model to evaluate the newly synthesized preference pairs and gather the judge's answer and reasoning traces. Since we know the preferred pair by construction, we retain only judgments that align with our synthetic preferences.
- **Judge model training.** We fine-tune the previous-iteration judge model on these filtered reasoning traces and answers.

We iterate this three-step process several times. We iteratively train a small Llama-3.2-11B model Grattafiori et al. (2024) using our framework, and evaluate the model with VL-RewardBench Li

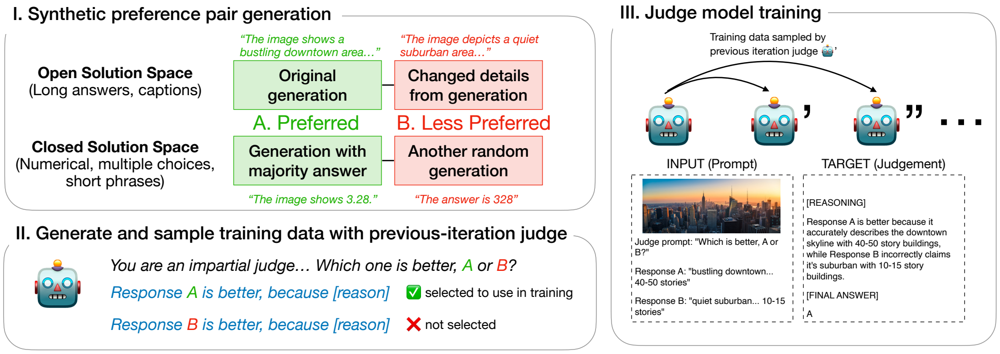

I. Synthetic preference pair generation
"The image shows a bustling downtown area..."
"The image depicts a quiet suburban area..."
Open Solution Space
(Long answers, captions)
Original generation
Changed details from generation
A. Preferred
B. Less Preferred
Closed Solution Space
(Numerical, multiple choices, short phrases)
Generation with majority answer
Another random generation
"The image shows 3.28."
"The answer is 328"
II. Generate and sample training data with previous-iteration judge
You are an impartial judge... Which one is better, A or B?
Response A is better, because [reason]
✅ selected to use in training
Response B is better, because [reason]
❌ not selected
III. Judge model training
Training data sampled by previous iteration judge 🤖'
' '' ...
INPUT (Prompt)
TARGET (Judgement)
Judge prompt: "Which is better, A or B?"
Response A: "bustling downtown... 40-50 stories"
Response B: "quiet suburban... 10-15 stories"
[REASONING]
Response A is better because it accurately describes the downtown skyline with 40-50 story buildings, while Response B incorrectly claims it's suburban with 10-15 story buildings.
[FINAL ANSWER]
A

Figure 1: Self-improving VLM judge: iterative synthetic preference data generation and judge model fine-tuning pipeline. **I. Synthetic preference pair generation.** We create synthetic preference pairs tailored to different question types. For open solution space questions (long answers, captions), we generate an original response and deliberately inject meaningful errors to create a less preferred version. For closed solution space questions (numerical, multiple choice, short phrases), we generate multiple candidates and pair the majority answer with a random alternative. **II. Iterative judge training data generation.** We use the previous-iteration judge model to evaluate the newly synthesized preference pairs and gather the judge's reasoning traces. We retain only judgments that align with our synthetic preferences. **III. Judge model training.** We fine-tune the previous-iteration judge model on these filtered reasoning traces. We iterate this three-step process several times. More details in §2.

et al. (2025) and Multimodal RewardBench Yasunaga et al. (2025). Our method improves the model from 0.38 to 0.51 overall accuracy, often outperforming much larger models like Llama-3.2-90B and Claude 3.5 Sonnet Anthropic (2024) on VL-RewardBench. In Section 5, we systematically analyze the impact of synthetic data design and iterative refinement, providing insights into the conditions under which self-improvement is most effective for VLM judge training. Specifically, our synthetic data construction provides a way to create a preference dataset from any visual queries, even without any reference answers. This makes our framework applicable to new visual tasks where ground-truth annotations are unavailable or scarce, such as evaluating responses to novel image collections or emerging visual domains. Overall, our contributions are:

- A method to create diverse synthetic preference data for VLMs without human preference annotations, using majority voting consensus for closed-ended tasks and controlled error injection for open-ended tasks.
- An iterative self-improvement framework that trains VLM judges by filtering and learning from self-generated reasoning traces, enabling iterative refinement of evaluation capabilities.
- Empirical demonstration that our approach enables a compact 11B model to gain substantial improvements on several reward benchmarks, even surpass much larger models (90B) and proprietary systems (Claude 3.5 Sonnet) on VL-RewardBench.
- Analysis showing scaling trend over increasing iterations, and operates effectively without access to ground-truth answers, making it applicable to emerging visual domains where annotations are scarce or unavailable.

2 Method

Our self-improving multimodal judge framework consists of three key stages: synthetic preference pair generation (Section 2.1), constructing judge training data with filtering (Section 2.2), and iterative training (Section 2.3). The key idea in this process is to generate diverse data that best help the model learn how to judge.

2.1 Synthetic Data Generation

The central challenge in training multimodal judges lies in obtaining high-quality preference pairs that reflect realistic failure modes of vision-language models. Human annotation is costly and does not scale across domains, while relying on stronger external models introduces undesirable dependencies. We instead ask: *can a model generate its own training signal by creating and* *recognizing flawed outputs?* Although self-improvement has been explored in text-only settings, extending it to multimodal evaluation is non-trivial: the model must detect errors that are intrinsically visual, such as object miscounting, incorrect spatial reasoning, or hallucinated image content.

Our approach is fully self-contained: the same model generates candidate responses and systematically perturbs them to induce controlled errors. This design ensures that learning is exploiting the model's internal knowledge rather than distilling from an external teacher. Crucially, the synthetic data construction emphasizes semantically meaningful contrasts instead of superficial edits. Because preference directions are known by construction, no human judgments are required.

We categorize evaluation tasks based on the degree of answer determinism. Tasks such as captioning or long-form reasoning yield *open-ended* responses without objectively verifiable solutions, whereas tasks with multiple-choice, numerical, or short-form answers yield closed-ended responses with deterministic correctness. Formally, let $\mathcal{D} = \{(I_i, Q_i)\}_{i=1}^{n}$ denote a collection of image-question pairs, where I_i represents the i-th image and Q_i the corresponding question. We denote the open-ended subset as $\mathcal{D}_{nv}$ and the closed-ended subset as $\mathcal{D}_v$.

2.1.1 Detail Alteration for Open-Ended Task

For captions and long-form reasoning responses, we construct preference pairs by introducing controlled inaccuracies into one response, producing a natural ranking without the need for absolute correctness. For each image–question pair, we sample two response sets from a base VLM. One is treated as the original response, while the other is modified by prompting the model to inject specific errors—such as altered counts or object attributes—while preserving style, structure, and length. These perturbations reflect documented VLM failure modes (Liu et al., 2024; Zhou et al., 2024a). The resulting responses remain contextually plausible but factually incorrect, e.g., referencing objects that could plausibly appear but are absent from the image. This targets a known VLM weakness: conflating visual plausibility with visual evidence (Liu et al., 2024; Zhou et al., 2024a). Consequently, the judge must rely on fine-grained visual grounding rather than linguistic priors. Prompts used for response alteration are listed in Appendix C.

For captioning tasks, we further increase diversity by varying prompt styles and length constraints, preventing reliance on a single stylistic mode. Prompts specifying short or long captions and different stylistic exemplars are randomly sampled; the full set is provided in Appendix D. Although absolute correctness cannot be enforced, systematic error injection yields reliable relative preferences. This contrastive setup trains the judge to detect targeted degradations rather than optimize for absolute accuracy, reducing amplification of base-model biases.

Formally, for each $(I, Q) \in \mathcal{D}_{nv}$, we generate two response sets using the base model M_{base}. One response is designated as the original $T = M_{\text{base}}(I, Q)$, while the other is altered by prompting M_{base} to introduce inaccuracies, producing T'. This yields preference pairs (I, Q, T, T').

Image	[City skyline]
Original Answer (T)	The image shows a bustling downtown area with modern skyscrapers reaching heights of approximately 40–50 stories. The glass facades reflect the afternoon sunlight.
Altered Answer ($\mathbf{T'}$)	The image shows a quiet suburban area with colonial buildings reaching heights of approximately 10–15 stories. The brick facades absorb the evening moonlight.

Table 1: Example of detail alteration for synthetic data generation

2.1.2 Majority Voting for Closed-Ended Tasks

For closed-ended tasks, altering descriptive details is ineffective, as errors stem from incorrect reasoning rather than surface-level content. Simply flipping short answers (e.g., from "A" to "D") yields trivial preference pairs with limited training signal. We therefore exploit the determinism of closed-ended tasks by using majority voting to identify likely correct responses and pairing them with randomly sampled alternatives.

For each $(I, Q) \in \mathcal{D}_v$, we generate N responses from the base model and select the majority answer as T^*. We then sample a different response $T^- \neq T^*$ to form preference pairs (I, Q, T^*, T^-). To ensure reliability, we retain only cases where at least five responses agree on T^*, discarding high-variance examples.

2.1.3 Correct Answer Filtering

As a comparison to majority voting, we also construct preference pairs using dataset-provided ground truth labels. Unlike majority voting, which derives supervision from model consistency, this approach relies on external correctness signals that may not align with what the model can reliably evaluate.

For each (I, Q) with gold label G, we generate N responses and construct preference pairs (I, Q, T^+, T^-) where T^+ matches G and T^- does not.

2.2 Training Data Generation

We sample the model's own reasoning traces only when its judgments align with our synthetic preference assumptions, inspired by self-training literature Wang et al. (2024); Hu et al. (2024). For each preference pair (I, Q, T^+, T^-), we query the current judge $M_{\text{judge}}^{(k)}$ at iteration k to generate N judgments:

$$\{J_1, J_2, \ldots, J_N\} \;=\; \{\, M_{\text{judge}}^{(k)}(I, Q, T^+, T^-)\,\}_{j=1}^{N}$$

Each judgment J_j contains a reasoning trace R_j and a binary decision $D_j \in \{0, 1\}$ indicating whether T^+ is preferred. We construct the training set $\mathcal{T}^{(k+1)}$ by retaining only correct judgments:

$$\mathcal{T}^{(k+1)} \;=\; \{(I, Q, T^+, T^-, R_j, 1) \;:\; D_j = 1\}$$

To mitigate positional bias, we evaluate each pair in both orderings (T^+, T^-) and (T^-, T^+), only samples where the judge is correct in both configurations. This ensures that retained reasoning reflects genuine preference discrimination rather than positional heuristics.

Intuitively, we bootstrap learning only from reasoning traces associated with correct judgments, reinforcing reliable evaluation patterns while avoiding error amplification.

2.3 Training

We train the judge via supervised fine-tuning on $\mathcal{T}^{(k+1)}$, maximizing the likelihood of producing correct reasoning traces and judgments:

$$\mathcal{L} = -\sum_{(I,Q,T^+,T^-,R,D)\in\mathcal{T}^{(k+1)}} \log P(R, D \mid I, Q, T^+, T^-)$$

The model input consists of the image, question, and candidate responses, while the output includes both the reasoning trace and final decision.

3 Experimental Setups

3.1 Dataset

We construct our synthetic training set from LLaVA-OneVision Li et al. (2024a), which includes multimodal tasks such as reasoning, math and coding, and captioning. To focus on multimodal evaluation, we restrict the dataset to single-image subsets. We cap each sub-dataset at 10k examples, resulting in 100k total prompts, to ensure balanced category coverage and manageable dataset size.

3.2 Evaluation

We evaluate our VLM judge and baselines on two benchmarks: Multimodal RewardBench Yasunaga et al. (2025) and VL-RewardBench Li et al. (2025). Both assess judge performance across general instruction following, hallucination detection, correctness, reasoning, and VQA. Multimodal RewardBench contains 5,211 preference pairs covering long-form and short-answer responses, labeled using ground-truth correctness or expert human annotations. VL-RewardBench focuses on general multimodal instructions, hallucination detection, and complex reasoning, combining human-verified and AI-annotated preferences. Together, these benchmarks provide standard protocols for evaluating alignment with human judgments.

	VLRB				*MMRB*					
Model	**Ave.**	**Gen.**	**Hallu.**	**Reason.**	**Ave.**	**Gen.**	**Know.**	**Reason.**	**Safe.**	**VQA**
Larger Models										
Claude-3.5	0.536	0.434	0.550	0.623	0.721	0.652	0.739	0.669	0.687	0.856
GPT-4o	0.624	0.491	0.676	0.705	0.713	0.658	0.720	0.649	0.668	0.872
Llama-90B	0.539	0.426	0.573	0.617	0.618	0.642	0.612	0.547	0.519	0.771
Our Method (Based on Llama-3.2-11B-Vision-Instruct)										
Base	0.383	0.297	0.304	0.549	0.499	0.590	0.506	0.517	0.317	0.565
Iteration 1	0.452	0.398	0.362	0.596	0.519	**0.625**	0.532	0.500	0.289	0.651
Iteration 2	0.521	0.453	**0.529**	0.580	0.521	0.591	0.506	0.513	0.304	0.693
Iteration 3	0.488	0.425	0.426	**0.612**	0.538	0.599	0.540	**0.543**	0.307	**0.701**
Iteration 4	**0.538**	**0.503**	0.514	0.596	**0.539**	0.591	**0.556**	0.531	**0.329**	0.689

Table 2: Performance comparison on VLRB and MMRB benchmarks. Our iteratively trained judge model based on Llama-3.2-11B achieves competitive performance with significantly larger models across multiple evaluation dimensions.

3.3 Implementation details

We fine-tune Llama-3.2-11B-Vision-Instruct for 5 epochs with a learning rate of $1\mathrm{e}{-5}$ and a batch size of 2 per GPU across 8 GPUs. Training uses FSDP, fast kernels, and cross-entropy loss over generated reasoning traces. Iterative training continues until performance plateaus, defined as less than 1% relative improvement over three consecutive iterations on either benchmark. Performance gains diminish after iteration 4, with dimension-specific peaks. Additional hyperparameters are provided in Appendix E.

4 Results

4.1 Iterative Training Improvements

Table 2 shows that iterative training substantially improves the Llama3-11B-based judge. On VLRB, performance increases from 0.383 at initialization to 0.538 after four iterations, corresponding to a 40.5% relative gain. On MMRB, performance improves from 0.499 to 0.539. Improvements are generally consistent across iterations, though the rate of progress varies by evaluation dimension.

Despite its smaller scale, the 11B judge achieves competitive or superior performance relative to larger models. On general instruction following, it outperforms Llama-3.2-90B (0.426), Claude-3.5-Sonnet (0.434), and GPT-4o (0.491), achieving a score of 0.503. On VLRB hallucination detection and MMRB VQA, the model (0.514 and 0.689) approaches the performance of these larger models despite a much weaker starting point (0.304 and 0.565).

Extended experiments through iteration 6 (Appendix G) show that gains are largest up to iteration 4, after which performance becomes dimension-dependent: some metrics plateau while others slightly degrade. This behavior aligns with expected convergence patterns in literature Wang et al. (2024).

To assess scalability, we also apply our method to Llama-3.2-90B-Vision-Instruct (Appendix H). The 90B judge improves from 0.618 to 0.690 on MMRB and from 0.539 to 0.628 on VLRB, confirming that our approach generalizes across model sizes.

4.2 Performance Gains Across Dimensions

Iterative training exhibits clear dimension-specific learning dynamics. On VLRB, general instruction following shows the largest relative gain of 69% (0.297 → 0.503), indicating effective supervision for open-ended multimodal evaluation. Hallucination detection improves by 40.9% (0.304 → 0.514), validating our detail-alteration strategy for training factual consistency. Reasoning accuracy increases by 8.6% (0.549 → 0.596 at iteration 4), but exhibits non-monotonic behavior, peaking at iteration 3 (0.612), suggesting mild overfitting in later iterations.

On MMRB, VQA shows substantial improvement (18.0% relative gain, 0.565 → 0.689), demonstrating effective transfer from majority-voting-based preference construction. In contrast, other dimensions show limited gains: reasoning improves marginally (0.517 → 0.531), general evaluation remains nearly flat (0.590 → 0.591), and safety shows minimal responsiveness (0.317 → 0.329). These results suggest a distributional mismatch between our synthetic data and the supervision required for these capabilities.

4.3 Additional ablations

Synthetic Data Creation. We compare our method with POVID Zhou et al. (2024a), a hallucination-focused SoTA method. While POVID yields early gains on hallucination metrics, its reasoning performance degrades across iterations (Appendix I.1). By iteration 4, POVID reaches 0.481 on VLRB average, compared to our 0.538, with notably worse reasoning performance (0.455 vs. 0.596). This gap likely reflects POVID's focus on hallucination, whereas our approach targets general-purpose judgment.

Human-Supervised Baseline. We compare against LLaVA-Critic-8B Xiong et al. (2025), trained with human preference annotations. Despite using no human supervision, our model outperforms LLaVA-Critic on VLRB overall (0.538 vs. 0.507), achieves substantially better hallucination detection (0.514 vs. 0.383), and attains comparable reasoning performance (0.596 vs. 0.591) (Appendix I.2).

Together, these ablations demonstrate that (1) general-purpose synthetic preference construction yields broader improvements than task-specific methods, and (2) annotation-free synthetic data can effectively replace costly human preference labels for training VLM judges.

5 Analysis and Discussion

Our results show that self-improvement provides a promising path for building multimodal judges without relying on costly human annotations or stronger teacher models. In this section, we reflect on the strengths and limitations of our approach.

5.1 Scaling number of iterations

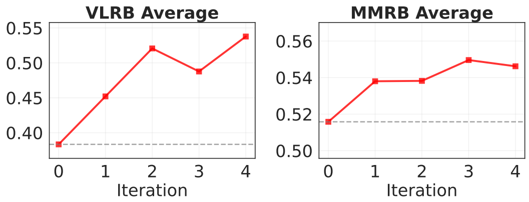


Figure 2: Judge model performance across training iterations. The left lanel shows average VLRB scores and the right panel shows average MMRB scores. After 4 iterations, our 11B judge model is comparable with Claude-3.5 and Llama-90B on VLRB.

Our iterative training scheme enables the judge to retain an increasing fraction of synthetic training pairs as performance improves. As shown in Figure 3, retained data increases from 19% at initialization to 43% by iteration 4, closely tracking performance gains. The largest improvement occurs in iteration 1 (10% average across VLRB and MMRB), followed by diminishing gains of approximately 3% per iteration thereafter. These trends mask dimension-specific behavior: VLRB General and Hallucination improve monotonically, while MMRB General Correctness exhibits non-monotonic fluctuations, indicating heterogeneous convergence rates across task domains.

Human Evaluation of Reasoning Traces. We evaluate whether reasoning quality improves across iterations via a blind human study on 80 randomly sampled examples using independent binary judgments. The fraction of reasoning traces rated as valid by both annotators increases from 33% at iteration 0 to 83% at iteration 4, a statistically significant improvement ($p < 0.01$, Cohen's $d = 1.18$, $\kappa = 0.58$). Example reasoning trajectories are provided in Appendix A.

5.2 Analysis on reasoning: filtering with majority votes v.s. correct answers

For reasoning and VQA tasks with short answers, we sample 16 responses and treat the majority answer as the preferred response, pairing it with a random alternative as the less preferred response.

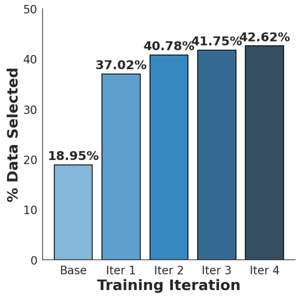


Figure 3: Increasing percentage of data sampled at each training iteration.

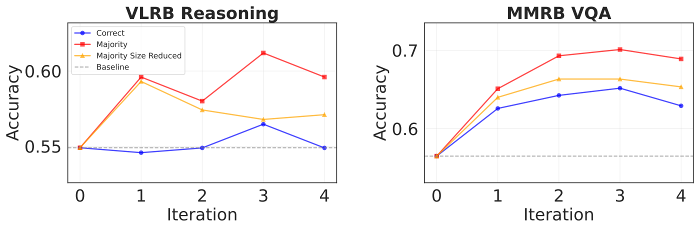


Figure 4: Majority voting vs. correctness filtering for synthetic pair selection. Majority voting yields higher performance on VLRB Reasoning and MMRB VQA; matching data size reduces but does not eliminate the gap.

The majority answer is not guaranteed to be correct, particularly on difficult examples. As an ablation, we also construct preference pairs using oracle correct answers from the dataset.

Figure 4 compares these strategies on closed-ended tasks. Majority-voted preference pairs outperform correctness-filtered pairs by 8.6% on VLRB-Reasoning and 9.5% on MMRB-VQA. We attribute this advantage to two factors. First, majority voting yields a larger number of retained training samples per iteration, providing a stronger optimization signal; constraining the majority-voted set to match the size of the correctness-filtered set reduces performance, but does not eliminate the gap. Second, correctness of the preferred answer does not guarantee valid reasoning: the reasoning may focus on stylistic features over factual accuracy, even when selecting the correct answer (Appendix 4).

Overall, while correctness filtering improves performance, majority voting is both more sample-efficient and more effective. Crucially, it removes dependence on ground-truth labels, allowing the method to scale to new image domains without annotated answers.

5.3 Analysis on other task dimensions

In §5.2, we shows the effectiveness of our method on reasoning and VQA tasks. In this section, we analyze the performance on other task dimensions. Specifically, we focus on the categories that exhibit the most and the least significant improvements.

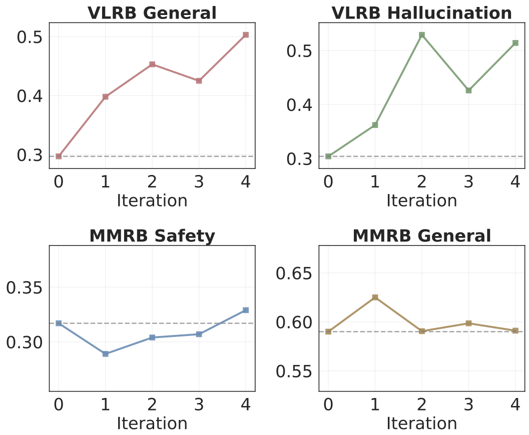


Figure 5: The dimensions that showed the most significant improvements (VLRB General, Hallucination) and the least significant improvements (MMRB Safety, General). More details in §5.3.

VLRB General & Hallucination. Our self-improving pipeline performs particularly well on VLRB General and Hallucination. The iteratively trained Llama-3.2-11B judge outperforms both Llama-3.2-90B and Claude-3.5-Sonnet on VLRB General, while also achieving substantial gains in hallucination detection. These results are notable given that VLRB General draws from WildVision Lu et al. (2024) and VLFeedback Li et al. (2024c), which reflect real-world human preferences and instruction-following behavior. The strong performance suggests that our synthetic data generation effectively captures supervision signals present in real-world VLM interactions.

MMRB General. Performance on MMRB General is relatively flat compared to other dimensions, likely due to increased image diversity. This category combines NoCaps Agrawal et al. (2019) and VisitBench Bitton et al. (2023). After four iterations, VisitBench shows clear gains (correctness 0.54 → 0.64), whereas NoCaps plateaus or degrades (0.58 → 0.52). Since NoCaps sources images from OpenImages Kuznetsova et al. (2020), which exhibits substantially greater visual and object diversity, this divergence suggests a domain mismatch between training and evaluation images. While our approach expands task diversity via LLaVA-OneVision Li et al. (2024a), incorporating more diverse images may be necessary to improve generalization on datasets.

MMRB Safety. Improvements on safety benchmarks are limited and inconsistent. This likely reflects the absence of safety-specific synthetic data, as our generation process does not explicitly target biased or harmful content. These results highlight the importance of task-aligned data synthesis: improving safety evaluation likely requires dedicated red-teaming and guardrail-focused data generation, which we leave for future work.

6 Related Work

LLM and VLM as a Judge. LLM-as-a-judge provides scalable reward signals for RL fine-tuning, reducing dependence on human annotations. In text-only settings, LLMs are widely used as automatic evaluators for summarization, dialogue, and reasoning, with strong agreement with human judgments. Prior work explores diverse training strategies for LLM-based judges, including AI feedback and self-rewarding approaches Zheng et al. (2023); Lee et al. (2023); Yuan et al. (2024); Cui et al. (2023); Lambert et al. (2024); Whitehouse et al. (2025); Saha et al. (2025). VLMs have similarly been used as judges for captioning, VQA, and reasoning, with most approaches scaling human preference collection or distilling supervision from large closed-source models Xiong et al. (2025); Wang et al. (2025); Li et al. (2025); Yasunaga et al. (2025). While some methods target specific dimensions (e.g., hallucination), general-purpose multimodal judges typically require substantial preference data Jing & Du (2024); Sun et al. (2023a).

Synthetic Data for Model Self-Improvement. Synthetic data drives self-improvement by enabling models to generate, evaluate, and refine their own outputs. Instruction tuning bootstraps new tasks from model-generated data, while iterative refinement supports self-critique and response improvement Wang et al. (2022); Li et al. (2024b); Madaan et al. (2023); Shinn et al. (2023). Self-rewarding and self-play methods show that synthetic supervision can outperform human-labeled baselines Alemohammad et al. (2024). In multimodal settings, scarce annotations increase reliance on synthetic data, but VLMs face additional challenges such as hallucination Sun et al. (2023b); Zhou et al. (2024a;c); Leng et al. (2023) and text-image misalignment Zhou et al. (2024b). Most prior work relies on larger models for data generation; our work avoids this by maintaining visual grounding and factual accuracy without human annotations or stronger teachers.

7 Conclusion

We show that strategic synthetic data generation with iterative refinement can train effective VLM judges without human preference annotations. Our 11B model surpasses the 90B Llama-3.2-90B baseline by 18% on VLRB (0.538 vs. 0.426 on general instructions), outperforms GPT-4o on instruction following, and approaches its hallucination detection performance (0.514 vs. 0.676). Starting from a weak base model (0.383 VLRB), four self-improvement iterations yield a 40.5% relative gain and transfer across model scales. The annotation-free framework readily extends to emerging modalities such as video understanding, multi-image reasoning, and 3D vision, where human preference labels are scarce. Remaining gaps in safety and reasoning highlight directions for targeted data synthesis.

References

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8948–8957, 2019.

Sina Alemohammad, Ahmed Imtiaz Humayun, Shruti Agarwal, John Collomosse, and Richard Baraniuk. Self-improving diffusion models with synthetic data. *arXiv preprint arXiv:2408.16333*, 2024.

Anthropic. Introducing the next generation of claude. `https://www.anthropic.com/news/claude-3-family`, March 2024.

Yonatan Bitton, Hritik Bansal, Jack Hessel, Rulin Shao, Wanrong Zhu, Anas Awadalla, Josh Gardner, Rohan Taori, and Ludwig Schmidt. Visit-bench: A benchmark for vision-language instruction following inspired by real-world use. *arXiv preprint arXiv:2308.06595*, 2023.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. In *Forty-first International Conference on Machine Learning*, 2024.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. Ultrafeedback: Boosting language models with scaled ai feedback. *arXiv preprint arXiv:2310.01377*, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yushi Hu, Otilia Stretcu, Chun-Ta Lu, Krishnamurthy Viswanathan, Kenji Hata, Enming Luo, Ranjay Krishna, and Ariel Fuxman. Visual program distillation: Distilling tools and programmatic reasoning into vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9590–9601, 2024.

Liqiang Jing and Xinya Du. Fgaif: Aligning large vision-language models with fine-grained ai feedback. *arXiv preprint arXiv:2404.05046*, 2024.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Harrison Lee, Sheng Shen, Mike Wu, David Ramirez, Serena Yeung, James Zou, and Dawn Song. Rlaif: Scaling reinforcement learning from ai feedback. *arXiv preprint arXiv:2309.00267*, 2023. URL `https://arxiv.org/abs/2309.00267`.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding, 2023. URL `https://arxiv.org/abs/2311.16922`.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer, 2024a. URL `https://arxiv.org/abs/2408.03326`.

Hao Li, Wei Liang, Yutong Chen, Shujian Zhang, Rui Wang, Ying Luo, and Jie Zhou. Generalized instruction tuning for language models. *arXiv preprint arXiv:2405.15972*, 2024b. URL `https://arxiv.org/abs/2405.15972`.

Lei Li, Zhihui Xie, Mukai Li, Shunian Chen, Peiyi Wang, Liang Chen, Yazheng Yang, Benyou Wang, Lingpeng Kong, and Qi Liu. Vlfeedback: A large-scale ai feedback dataset for large vision-language models alignment. *arXiv preprint arXiv:2410.09421*, 2024c.

Lei Li, Yuancheng Wei, Zhihui Xie, Xuqing Yang, Yifan Song, Peiyi Wang, Chenxin An, Tianyu Liu, Sujian Li, Bill Yuchen Lin, Lingpeng Kong, and Qi Liu. Vl-rewardbench: A challenging benchmark for vision-language generative reward models, 2025. URL `https://arxiv.org/abs/2411.17451`.

Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Mitigating hallucination in large multi-modal models via robust instruction tuning, 2024. URL `https://arxiv.org/abs/2306.14565`.

Yujie Lu, Dongfu Jiang, Wenhu Chen, William Yang Wang, Yejin Choi, and Bill Yuchen Lin. Wildvision: Evaluating vision-language models in the wild with human preferences. *Advances in Neural Information Processing Systems*, 37:48224–48255, 2024.

Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023. URL `https://arxiv.org/abs/2303.17651`.

Swarnadeep Saha, Xian Li, Marjan Ghazvininejad, Jason Weston, and Tianlu Wang. Learning to plan reason for evaluation with thinking-llm-as-a-judge, 2025. URL `https://arxiv.org/abs/2501.18099`.

Noah Shinn, Federico Cassano, Yonatan Bisk Labash, Kurt Shuster, Jason Weston, and Jing Liu. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL `https://arxiv.org/abs/2303.11366`.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. Aligning large multimodal models with factually augmented rlhf. 2023a.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. Aligning large multimodal models with factually augmented rlhf, 2023b. URL `https://arxiv.org/abs/2309.14525`.

Tianlu Wang, Ilia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. Self-taught evaluators, 2024. URL `https://arxiv.org/abs/2408.02666`.

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *arXiv:2212.10560*, 2022. URL `https://arxiv.org/abs/2212.10560`.

Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. J1: Incentivizing thinking in llm-as-a-judge via reinforcement learning, 2025. URL `https://arxiv.org/abs/2505.10320`.

Tianyi Xiong, Xiyao Wang, Dong Guo, Qinghao Ye, Haoqi Fan, Quanquan Gu, Heng Huang, and Chunyuan Li. Llava-critic: Learning to evaluate multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13618–13628, June 2025.

Michihiro Yasunaga, Luke Zettlemoyer, and Marjan Ghazvininejad. Multimodal rewardbench: Holistic evaluation of reward models for vision language models, 2025. URL `https://arxiv.org/abs/2502.14191`.

Tianyu Yu, Haoye Zhang, Qiming Li, Qixin Xu, Yuan Yao, Da Chen, Xiaoman Lu, Ganqu Cui, Yunkai Dang, Taiwen He, et al. Rlaif-v: Open-source ai feedback leads to super gpt-4v trustworthiness. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19985–19995, 2025.

Weizhe Yuan, Richard Yuanzhe Pang, et al. Self-rewarding language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, 2024. URL `https://arxiv.org/abs/2401.10020`.

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*, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *NeurIPS 2023*, 2023. URL `https://arxiv.org/abs/2306.05685`.

Yiyang Zhou, Chenhang Cui, Rafael Rafailov, Chelsea Finn, and Huaxiu Yao. Aligning modalities in vision large language models via preference fine-tuning, 2024a. URL `https://arxiv.org/abs/2402.11411`.

Yiyang Zhou, Chenhang Cui, Rafael Rafailov, Chelsea Finn, and Huaxiu Yao. Aligning modalities in vision large language models via preference fine-tuning, 2024b. URL `https://arxiv.org/abs/2402.11411`.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. Analyzing and mitigating object hallucination in large vision-language models, 2024c. URL `https://arxiv.org/abs/2310.00754`.

A TRAINING EXAMPLE FROM DIFFERENT ITERATIONS

Component	Content
Response A	The image shows a blue double-decker bus driving down a street in a town or city. The bus is in the center of the image and has a large front window and a sign on the top that reads Brighton Open Top in yellow letters. The bus is driving away from the camera, and there are people standing on the sidewalk next to it. In the background, there are several buildings lining the street, including a row of shops and a few houses. The sky is overcast, and the overall atmosphere of the image is one of a quiet, everyday scene in a small town or city.
Response B	The image shows a red double-decker bus driving down a street in a town, surrounded by tall buildings and parked cars. The bus has a red color and features a large front window, with the words ”Brighton Open Top” written in yellow on the front. The bus appears to be in motion, with the front wheels turned to the left as it navigates the street. In the background, the buildings are tall and white, with many windows and balconies. Some of the buildings have signs or advertisements on them, but they are not clearly visible. The sky above is blue and cloudy, adding to the overall atmosphere of the scene. Overall, the image captures a typical urban scene, with the red bus being the main focus of attention. The surrounding buildings and parked cars add depth and context to the image, while the cloudy sky provides a sense of atmosphere and mood.
Iteration 1 Reasoning	Both responses aim to describe the image, but they differ in their level of detail, accuracy, and relevance to the user’s question. Assistant A’s response provides a clear and concise description, accurately identifies the bus as blue, and notes people on the sidewalk. Assistant B’s response provides more detailed description including the bus’s color (red) and parked cars, and describes the surrounding buildings’ height and signs. While both provide accurate information, Assistant B’s response is more detailed and engaging, making it a better choice.
Iteration 2 Reasoning	The two responses have similarities but notable differences. Assistant A’s response is more detailed and descriptive, highlighting the red double-decker bus, tall buildings, and cloudy sky with simple, accessible language. Assistant B’s response is more concise and lacks detail, only mentioning the blue double-decker bus and surrounding buildings without additional context.
Iteration 3 Reasoning	The two responses differ significantly in accuracy, relevance, and depth. Assistant A’s response is more accurate and relevant, correctly identifying the bus as red and describing features like the large front window and ”Brighton Open Top” text. It provides detailed description of surrounding buildings and parked cars, adding context. Assistant B’s response is less accurate, incorrectly identifying the bus as blue and failing to mention the red color. It lacks detailed description of surroundings, making it less informative.

Table 3: Comparison of assistant responses and judge reasoning across training iterations for the double-decker bus image example. Response A describes a blue bus while Response B describes a red bus. The judge’s reasoning improves from iteration 1 (focusing on detail and engagement) to iteration 3 (explicitly identifying factual accuracy errors).

B JUDGE PROMPT

```
Please act as an impartial judge and evaluate the quality of the
responses provided by two AI assistants to the user question displayed
below. You should choose the assistant that follows the user’s
instructions and answers the user’s question better. Your evaluation
should consider factors such as the helpfulness, relevance, accuracy,
depth, creativity, and level of detail of their responses. Begin your
evaluation by comparing the two responses and provide a short
explanation. Avoid any position biases and ensure that the order in
which the responses were presented does not influence your decision.
Do not allow the length of the responses to influence your evaluation.
Do not favor certain names of the assistants. Be as objective as
possible. After providing your explanation, output your final verdict
by strictly following this format: ’[[A]]’ if assistant A is better,
’[[B]]’ if assistant B is better.

[User Question]
{q}
```

```
[The Start of Assistant A’s Answer]
{r1}
[The End of Assistant A’s Answer]

[The Start of Assistant B’s Answer]
{r2}
[The End of Assistant B’s Answer]

YOU MUST STRICTLY FOLLOW THE FORMAT BELOW. Start with **EVALUATION**
and make sure to add double brackets around the final answer.

**EVALUATION**: Provide a detailed comparison of both responses,
analyzing their strengths and weaknesses based on the above factors.
Be specific about why one response better serves the user’s needs.

**FINAL ANSWER**: ’[[A]]’ if assistant A is better, ’[[B]]’ if
assistant B is better.
```

C Detail Alternation Prompts

```
INSTRUCTION="Below is a text (description, reasoning, explanation, etc.).
Your task is to generate a version that incorporates alternative details
while keeping original style and adhering to the following guidelines:
1. Detail Modifications: Replace exactly TWO specific details (e.g., visual
elements, facts, examples, reasoning steps, methods, premises) with
alternative details, or change the perspective on TWO features. 2. Length,
Structure and Style: Keep the overall length, structure, and style identical
to the original. 3. Core Identity & Function: Ensure the main subject’s
identity and function remain unchanged. 4. Tone & Detail Level: Maintain the
original tone and level of detail. 5. Style Mimicry: Mirror the exact writing
style, vocabulary choices, sentence patterns, and linguistic quirks of the
original text.; Below is a text (description, reasoning, explanation, etc.).
Your task is to generate a reinterpreted version that incorporates ONE
alternative detail while adhering to the following guidelines: 1. Detail
Modifications: Replace exactly ONE specific detail (e.g., visual element,
fact, example, reasoning step, method, premise) with alternative details, or
change the perspective on ONE feature. 2. Length, Structure and Style: Keep
the overall length, structure, and style identical to the original. 3. Core
Identity & Function: Ensure the main subject’s identity and function remain
unchanged. 4. Tone & Detail Level: Maintain the original tone and level of
detail. 5. Style Mimicry: Mirror the exact writing style, vocabulary choices,
sentence patterns, and linguistic quirks of the original text.; Below is a
text (description, reasoning, explanation, etc.). Your task is to generate a
reinterpreted version that incorporates TWO alternative details while
adhering to the following guidelines: 1. Detail Modifications: Replace
exactly TWO specific details (e.g., visual elements, facts, examples,
reasoning steps, methods, premises) with alternative details, or change the
perspective on TWO features. 2. Length, Structure and Style: Keep the overall
length, structure, and style identical to the original. 3. Core Identity &
Function: Ensure the main subject’s identity and function remain unchanged.
4. Tone & Detail Level: Maintain the original tone and level of detail.
5. Style Mimicry: Mirror the exact writing style, vocabulary choices,
sentence patterns, and linguistic quirks of the original text.; Below is a
text (description, reasoning, explanation, etc.). Your task is to generate a
reinterpreted version that incorporates 1-2 alternative details while
adhering to the following guidelines: 1. Detail Modifications: Replace
exactly 1-2 specific details (e.g., visual elements, facts, examples,
reasoning steps, methods, premises) with alternative details, or change the
perspective on 1-2 features. 2. Length, Structure and Style: Keep the overall
length, structure, and style identical to the original. 3. Core Identity &
Function: Ensure the main subject’s identity and function remain unchanged.
4. Tone & Detail Level: Maintain the original tone and level of detail.
5. You MUST make the required modifications at the beginning of the text.
5. Style Mimicry: Mirror the exact writing style, vocabulary choices,
sentence patterns, and linguistic quirks of the original text.; Below is a
text (description, reasoning, explanation, etc.). Your task is to generate a
reinterpreted version that incorporates 1-2 alternative details while
adhering to the following guidelines: 1. Detail Modifications: Replace
exactly 1-2 specific details (e.g., visual elements, facts, examples,
reasoning steps, methods, premises) with alternative details, or change the
perspective on 1-2 features. 2. Length, Structure and Style: Keep the overall
length, structure, and style identical to the original. 4. Tone & Detail
Level: Maintain the original tone and level of detail. 5. You MUST make the
required modifications close to the END of the text. 5. Style Mimicry: Mirror
the exact writing style, vocabulary choices, sentence patterns, and
linguistic quirks of the original text.; Below is a text (description,
reasoning, explanation, etc.). Your task is to generate a reinterpreted
version that incorporates 1-2 alternative details while adhering to the
following guidelines: 1. Detail Modifications: Replace exactly 1-2 specific
details (e.g., visual elements, facts, examples, reasoning steps, methods,
premises) with alternative details, or change the perspective on 1-2
features. 2. Length, Structure and Style: Keep the overall length, structure,
and style identical to the original. 3. Core Identity & Function: Ensure the
main subject’s identity and function remain unchanged. 4. Tone & Detail
Level: Maintain the original tone and level of detail. 5. You MUST make the
required modifications close to the MIDDLE of the text, instead of at the
beginning or the end. 5. Style Mimicry: Mirror the exact writing style,
vocabulary choices, sentence patterns, and linguistic quirks of the original
text."
```

D GENERATION META PROMPT FOR CAPTIONS

```
{
"llava_image_description_prompts": {
    "concise": [
        "Describe the image concisely.",
        "Provide a brief description of the given image.",
        "Offer a succinct explanation of the picture presented.",
        "Summarize the visual content of the image.",
        "Give a short and clear explanation of the subsequent image.",
        "Share a concise interpretation of the image provided.",
        "Present a compact description of the photo’s key features.",
        "Relay a brief, clear account of the picture shown.",
        "Render a clear and concise summary of the photo.",
        "Write a terse but informative summary of the picture.",
        "Create a compact narrative representing the image presented."
    ],
    "detailed": [
        "Describe the following image in detail",
        "Provide a detailed description of the given image",
        "Give an elaborate explanation of the image you see",
        "Share a comprehensive rundown of the presented image",
        "Offer a thorough analysis of the image",
        "Explain the various aspects of the image before you",
        "Clarify the contents of the displayed image with great detail",
            "Characterize the image using a well-detailed description",
            "Break down the elements of the image in a detailed manner",
            "Walk through the important details of the image",
            "Portray the image with a rich, descriptive narrative",
            "Narrate the contents of the image with precision",
            "Analyze the image in a comprehensive and detailed manner",
            "Illustrate the image through a descriptive explanation",
            "Examine the image closely and share its details",
            "Write an exhaustive depiction of the given image"
        ]
    }
}
```

```
{
    "caption_styles": {
        "concise":{
            "chatgpt": "The image shows a golden retriever sitting
in a sunny park with green grass and trees in the background.
The dog appears happy and is looking directly at the camera
with its tongue out. There are some fallen leaves scattered
around, suggesting it might be autumn. The lighting is natural
and warm, creating a pleasant, cheerful atmosphere.",
            "claude": "A golden retriever sits attentively in a
park setting, its mouth open in what appears to be a content
pant. The warm lighting and scattered autumn leaves create a
peaceful scene, with the dog positioned centrally against a
backdrop of grass and mature trees. The composition suggests a
casual outdoor moment captured during a walk or play session.",
            "qwen": "Golden retriever dog sitting on grass in park
environment. Autumn season indicated by fallen leaves on ground.
Natural daylight illumination. Dog exhibits typical friendly
expression with open mouth. Background contains trees and
vegetation. Image quality appears high resolution with good
color saturation.",
            "llama": "This image depicts a golden retriever in an
outdoor setting. The dog is positioned on grass with trees
visible in the background. The scene appears to be during
daytime with natural lighting. The dog’s posture suggests it
is alert and engaged. The presence of fallen leaves indicates
the photograph was likely taken during autumn months.",
            "gemini": "A beautiful golden retriever enjoys a moment
in the park! The friendly pup is sitting on the grass,
surrounded by the warm colors of fall. With its tongue happily
hanging out, this good boy seems to be having a wonderful time
outdoors. The natural lighting and peaceful setting make for a
heartwarming scene that captures the joy of spending time with
our four-legged friends."
        }
    }
}
```

E HYPERPARAMETERS

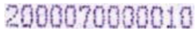
In data generation, we use a temperature of 0.6, top_k=50, top_p=0.9.

In data sampling for majority voting (Section 2), we set N = 16 to generate 16 candidate responses per closed-ended question. We experimented with different majority thresholds and found that requiring at least 5 identical responses (out of 16) provides the best balance between data quality and quantity. Lower thresholds (e.g., 3) produce more training data but include less reliable majorities, while higher thresholds (e.g., 8) overly restrict the dataset. We selected the threshold of 5 after empirically comparing thresholds of 3 and 5: both achieved similar final performance, but the threshold of 5 required less training data, suggesting better sample efficiency and filtering quality. This threshold effectively filters out cases with greater response diversity where the model's uncertainty is higher, retaining only examples where the model shows reasonable consensus.

In training, we use the default training setup in llama-cookbook which has a learning rate of 1e-5, we train for 5 epochs.

The data generation and model training were conducted on H100 GPUs, with approximately 400 GPUs hours. This represents a cost-efficient alternative to human annotation ($500K-1M for equivalent data) or GPT-4 distillation ($5K-10K in API costs).

F Correctness Filter Negative Example



2000070000010

Component	Content
Response A	The image contains a series of numbers, specifically “20000700000010”.
Response B	The image contains the text “2000070000010” in a purple, pixelated font. The text is centered at the top of the image and spans the entire width. It appears to be a numerical code or identifier, but without more context, it’s difficult to determine its meaning or significance.
Evaluation	Both responses attempt to decipher the content of the image, but they differ in their approach and level of detail. Assistant A’s response is concise and straightforward, simply stating the numbers visible in the image. However, it lacks context and fails to provide any additional information about the meaning or significance of these numbers. Assistant B’s response provides more detailed observations, including information about the font style, color, and layout of the text. It also acknowledges the limitations of interpretation without additional context. **FINAL ANSWER**: [[B]]

Table 4: Example of judge getting the correct answer but providing incorrect reasoning. Even though the final answer is correct, the reasoning failed to point out that the number in Response A has an extra 0 and it is a wrong answer.

While our filtering mechanism successfully removes many incorrect judgments, we acknowledge it is not perfect. This section illustrates a case where correctness- based filtering can retain problematic examples: the judge selects the correct answer (Response B) but fails to identify the specific transcription error in Response A (an extra digit ”20000700000010” vs. ”2000070000010”). Instead, the reasoning focuses on superficial features—Response B’s additional descriptive details and contextual observations—rather than the factual discrepancy that should determine the judgment.

This example suggests why filtering based on ground-truth correctness alone may be insufficient–even when the final judgment is correct, the reasoning may prioritize stylistic features over factual accuracy. This observation partially motivates our majority voting approach (Section 5.2), which filters training data based on the model’s own consistency rather than relying solely on external correctness labels. We hypothesize that by requiring the judge to correctly identify preferences across multiple synthetically generated pairs with diverse error types, majority voting may implicitly filter for more robust reasoning patterns rather than spurious correlations with superficial features. A judgment that succeeds consistently across varied synthetic contrasts could provide stronger evidence of genuine understanding than a single correct answer that might result from incidental heuristics.

We include this example not as definitive evidence of a fundamental flaw, but as an illustrative case that helps explain our design rationale. While we cannot conclusively verify that majority voting eliminates all imperfect reasoning, the empirical advantages shown in Section 5.2 suggest that consistency-based filtering may provide more robust supervision than correctness checking alone for learning generalizable judgment criteria.

G Extended Iterative Training Results

	VLRB				*MMRB*					
Model	**Ave.**	**Gen.**	**Hallu.**	**Reason.**	**Ave.**	**Gen.**	**Know.**	**Reason.**	**Safe.**	**VQA**
Larger Models										
Claude-3.5	0.536	0.434	0.550	0.623	0.721	0.652	0.739	0.669	0.687	0.856
GPT-4o	0.624	0.491	0.676	0.705	0.713	0.658	0.720	0.649	0.668	0.872
Llama-90B	0.539	0.426	0.573	0.617	0.618	0.642	0.612	0.547	0.519	0.771
Our Method (Based on Llama-3.2-11B-Vision-Instruct)										
Base	0.383	0.297	0.304	0.549	0.499	0.590	0.506	0.517	0.317	0.565
Iteration 1	0.452	0.398	0.362	0.596	0.519	**0.625**	0.532	0.500	0.289	0.651
Iteration 2	0.521	0.453	**0.529**	0.580	0.521	0.591	0.506	0.513	0.304	0.693
Iteration 3	0.488	0.425	0.426	**0.612**	0.538	0.599	0.540	0.543	0.307	**0.701**
Iteration 4	**0.538**	0.503	0.514	0.596	**0.539**	0.591	**0.556**	0.531	0.329	0.689
Iteration 5	0.519	**0.514**	0.482	0.562	0.535	0.591	0.548	**0.549**	0.332	0.653
Iteration 6	0.529	0.492	0.487	0.609	0.534	0.597	0.552	0.524	**0.352**	0.646

Table 5: Performance comparison on VLRB and MMRB benchmarks. Our iteratively trained judge model based on Llama-3.2-11B achieves competitive performance with significantly larger models across multiple evaluation dimensions.

H Scaling to larger model size

	MMRB						*VLRB*			
Model	**Ave.**	**Gen.**	**Know.**	**Reas.**	**Safe.**	**VQA**	**Ave.**	**Gen.**	**Hallu.**	**Reas.**
Base Llama-90B	0.618	0.642	0.612	0.547	0.519	0.771	0.539	0.426	0.573	0.617
Iteration 1	0.663	0.670	0.662	0.634	0.532	0.817	0.607	0.482	0.634	0.705
Iteration 2	0.682	0.683	0.694	0.666	**0.550**	0.817	0.616	0.503	**0.636**	0.715
Iteration 3	0.665	0.678	0.678	0.679	0.508	0.781	0.621	0.515	0.634	0.715
Iteration 4	**0.690**	**0.692**	**0.702**	**0.679**	0.550	**0.828**	**0.628**	**0.525**	0.634	**0.725**

Table 6: Llama-3.2-90B-Vision-Instruct performance across iterations. Despite different model capacity and systematic biases compared to the 11B model, we observe consistent improvement patterns (11.6% relative gain on MMRB, 16.5% on VLRB), validating the generalizability of our iterative training approach.

I Ablation Studies

I.1 Comparison with POVID-based Data Generation

Table 7 compares our synthetic data generation method with POVID Zhou et al. (2024a), a prior state-of-the-art approach specialized for hallucination detection. While POVID shows improvements on general and hallucination metrics in early iterations, it exhibits degrading performance on reasoning tasks across iterations, demonstrating the limitations of task-specific synthesis methods for building general-purpose judges.

	MMRB						*VLRB*			
Model	**Ave.**	**Gen.**	**Know.**	**Reas.**	**Safe.**	**VQA**	**Ave.**	**Gen.**	**Hallu.**	**Reas.**
Base Llama-11B	0.499	0.590	0.506	0.517	0.317	0.565	0.383	0.297	0.304	0.549
POVID Iter 1	0.512	0.622	0.524	0.528	0.387	0.498	0.420	0.354	0.358	0.549
POVID Iter 2	0.501	0.617	0.556	0.512	0.372	0.447	0.424	0.315	0.450	0.508
POVID Iter 3	0.480	0.592	0.505	0.486	0.353	0.466	0.493	0.514	0.467	0.498
POVID Iter 4	0.466	0.581	0.521	0.483	0.347	0.398	0.481	0.497	0.492	0.455
POVID Iter 5	0.470	0.584	0.498	0.502	0.355	0.410	0.487	0.506	0.497	0.458

Table 7: POVID-based data generation results across iterations. While POVID achieves peak hallucination performance at iteration 4 (0.492 on VLRB), it shows degrading reasoning performance (0.549 → 0.458) and inconsistent improvements across dimensions, contrasting with our method's stable improvements shown in Table 2.

I.2 Comparison with Human-Supervised Baseline

Table 8 compares our annotation-free method with LLaVA-Critic-8B **?**, a model trained with costly human-annotated preference pairs. Despite using no human annotations, our method achieves competitive or superior performance, validating the viability of purely synthetic training data for building effective multimodal judges.

Model	**Average**	**General**	**Hallucinations**	**Reasoning**
LLaVA-Critic-8B (human labels)	0.507	**0.546**	0.383	0.591
Ours (11B, no human labels)	**0.538**	0.503	**0.514**	**0.596**
Absolute Difference	*+0.031*	*-0.043*	*+0.131*	*+0.005*

Table 8: Comparison with LLaVA-Critic-8B on VLRB benchmark. Our annotation-free method outperforms the human-supervised baseline overall, with particularly strong gains on hallucination detection (+13.1 percentage points) while maintaining comparable reasoning performance.