VAPR – Vision-language Preference alignment for Reasoning

Rohan Wadhawan[1]*, Fabrice Harel-Canada[1], Zi-Yi Dou[1], Suhaila Shakiah[2], Robinson Piramuthum[2], Nanyun Peng[1]
[1]Department of Computer Science, University of California Los Angeles, USA
[2]Amazon.com, Inc., USA

Abstract

Preference finetuning methods like Direct Preference Optimization (DPO) with AI-generated feedback have shown promise in aligning Large Vision-Language Models (LVLMs) with human preferences. However, existing techniques overlook the prevalence of noise in synthetic preference annotations in the form of stylistic and length biases. To this end, we introduce a hard-negative response generation framework based on LLM-guided response editing, that produces rejected responses with targeted errors, maintaining stylistic and length similarity to the accepted ones. Using this framework, we develop the VAPR dataset, comprising 30K high-quality samples, to finetune three LVLM families: LLaVA-V1.5, Qwen2VL & Qwen2.5VL (2B-13B sizes). Our VAPR models deliver significant performance improvements across ten benchmarks, achieving average gains of 6.5% (LLaVA), 4.0% (Qwen2VL), and 1.5% (Qwen2.5VL), with notable improvements on reasoning tasks. A scaling analysis shows that performance consistently improves with data size, with LLaVA models benefiting even at smaller scales. Moreover, VAPR reduces the tendency to answer "Yes" in binary questions - addressing a common failure mode in LVLMs like LLaVA. Lastly, we show that the framework generalizes to open-source LLMs as editors, with models trained on VAPR-OS achieving 99% of the performance of models trained on VAPR, which is synthesized using GPT-4o. Our data, models, and code can be found on the project page https://vap-r.github.io/

1 Introduction

Recent advances in Large Vision-Language Models (LVLMs) have greatly enhanced their ability to perceive, reason, and generate open-ended responses to instructions (Liu et al., 2023c;b; 2024; Ye et al., 2024; Dai et al., 2023; Awadalla et al., 2023; Tong et al., 2024; Chen et al., 2024b; Team, 2024a; Chen et al., 2023; Zhu et al., 2023; Wang et al., 2024a). Despite the development, LVLMs often face challenges in vision-language alignment and reasoning, resulting in linguistically plausible texts that either contradict the visual context or lack logical reasoning. Several studies (Sun et al., 2023; Yu et al., 2024a;b; Zhou et al., 2024a;b; Deng et al., 2024; Li et al., 2023c; Zhao et al., 2023; Lee et al., 2023b; Sarkar et al., 2024) have focused on improving the modality alignment capabilities of instruction-tuned LVLMs through preference fine-tuning methods, including reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022; Schulman et al., 2017), AI feedback (RLAIF) (Lee et al., 2023a), and direct preference optimization (DPO) (Rafailov et al., 2024).

Early preference finetuning methods for LVLMs primarily incorporated human feedback (Sun et al., 2023; Yu et al., 2024a), but the high costs and limited scalability of this approach have led to a growing use of AI-generated feedback and synthetic preference dataset creation (Zhou et al., 2024a; Deng et al., 2024; Yu et al., 2024b; Zhou et al., 2024b; Lee et al., 2023b; Zhao et al., 2023). These synthetic preference datasets have demonstrated advantages over human-annotated versions, particularly in maintaining truthfulness (Ivison et al., 2024b) and enhancing modality alignment (Zhou et al., 2024a; Deng et al., 2024; Yu et al., 2024b;

*Correspondance to: rohanwadhawan7@gmail.com

Zhou et al., 2024b; Lee et al., 2023b; Zhao et al., 2023). However, our analysis reveals that these methods often overlook inadvertently injected "noises", including biases in length and style (Zhao et al., 2023) (see Table 1). As DPO has been found to exploit both length (Park et al., 2024; Lu et al., 2024) and stylistic biases (Yu et al., 2024a; Zhao et al., 2023; Hong et al., 2024; Yu et al., 2024b), this noise in the dataset can consequently undermine the alignment process.

To improve LVLMs' alignment and reasoning, we propose a hard-negative preference data generation framework and create VAPR, a dataset of 30K high-quality samples derived from the LLaVA-665K SFT dataset (Liu et al., 2023b). Each preference sample in VAPR pairs a ground truth response with a generated hard-negative response that preserves style and length while introducing targeted perturbations to ensure deliberate misalignment (see Fig. 1). While existing approaches (Zhou et al., 2024a; Wang et al., 2024b; Zhou et al., 2024b; Yu et al., 2024b) rely on VLMs (the same or a large oracle VLM) to generate and/or score sampled responses, we frame hard-negative response generation as ground truth response editing using an LLM - implemented via constrained data generation. This editing approach leverages LLMs' semantic understanding of text, which is typically more reliable than the image-text comprehension of VLMs (Guan et al., 2024). We provide the LLM with the instruction, ground truth response, and VL task-specific information (e.g., spatial reasoning, object attributes), and prompt it to perturb specific spans of the ground truth response. Unlike prior works, we use task-specific information to guide the editor in minimally editing the ground-truth response, injecting semantic errors that make the rejected response incorrect for the task while preserving style and length (see Table 1). This prevents models from exploiting superficial cues during preference optimization (§4.2, & §4.3.1). We also ensure that the VAPR corpus covers diverse instructions - perception (e.g., object existence, attribute, size, color, environment), reasoning (e.g., counting, spatial, comparison), and captioning - to help models learn generalizable preferences. For our main results, we use GPT-4o (OpenAI, 2024) as the LLM editor and refer to the approach as VAPR. Lastly, to evaluate the generalizability of the pipeline across different LLMs, we also conduct an ablation experiment using Qwen3-32b (Yang et al., 2025) as the editor, referred to as VAPR-OS (open-source), to distinguish it from the GPT-4o–based VAPR dataset.

Instruction: How many planes are visible in the image?
Chosen Response: There are four planes visible in the image.
VAPR Response: There are six planes visible in the image.
Response with length bias: There are seven planes visible in the sky in the image, each leaving a bright white trail behind.
Response with style bias: Five planes can be seen in the image.

Instruction: Where is the man standing in relation to the baby elephant?
Chosen Response: The man is standing next to the baby elephant in the water.
VAPR Response: The man is standing far away from the baby elephant, near the edge of the water.
Response with length bias: The man is standing far from the baby elephant, attempting to throw water using a bucket while standing at the edge of the water.
Response with style bias: The elephant is standing in the water, away from the man, who is on the shore.

Figure 1: Examples from the VAPR hard-negative generation framework show instruction, image, chosen response, and three rejected variants. (a) fine-grained perception and counting capability, and (b) spatial reasoning. VAPR introduces targeted error - modifying only task-relevant spans - while length-biased rejections add verbose description, and style-biased ones alter style and content. Blue highlights relevant spans in chosen response, green shows VAPR perturbations, and red indicates stylistic or length-biased edits.

To assess the effectiveness of the proposed framework, we fine-tune models from the LLaVA-V1.5-Instruct(7B, 13B), Qwen2VL-Instruct (2B, 7B), and Qwen2.5VL-Instruct (3B, 7B) families on the VAPR dataset and evaluate them across ten diverse benchmarks. These span open-

ended instruction following, vision-centric reasoning, academic and mathematical tasks, hallucination detection, and adversarial robustness. VAPR models outperform baselines on 8 out of 10 benchmarks, with average gains of 6.5% for LLaVA-V1.5, 4% for Qwen2VL, and 1.5% for Qwen2.5VL. Notably, VAPR yields consistent improvements on comprehensive benchmarks like SEED (Li et al., 2023a) and MMStar (Chen et al., 2024a), vision-centric reasoning benchmarks such as CV-Bench (Tong et al., 2024) (counting and spatial reasoning), and adversarial reasoning benchmarks like NaturalBench (Li et al., 2024a), which test compositional Visio-linguistic reasoning. Interestingly, improvements are also observed in textual and math reasoning despite no explicit training on such data. A scaling analysis reveals that performance scales with dataset size, with LLaVA-v1.5 models benefiting even with small preference tuning data budget, and Qwen2VL & Qwen2.5VL variants improving at larger scales.

We further compare VAPR to other preference datasets and empirically analyze how they function under DPO optimization. Our findings highlight the influence of stylistic and length differences in preference data, which can lead to spurious learning signals during training. VAPR mitigates this issue by generating hard-negative rejected responses, which are stylistically and length-wise similar to the chosen ones. We also observe that VAPR reduces the tendency to answer "Yes" in binary questions, addressing a common issue in LVLMs like LLaVA (Li et al., 2023d; Liu et al., 2023a; Guan et al., 2024; Li et al., 2024a). Lastly, our ablation using Qwen3-32b to generate VAPR-OS show that open-source models can follow the same prompting strategy, produce targeted perturbations, and yield models that outperform the base instruct model and achieve 99% performance of the VAPR models.

The contributions of this work are fourfold:

- We propose VAPR, a hard-negative generation framework based on LLM-guided response editing that constructs synthetic preference data with reduced stylistic and length biases. Using this framework, we create a 30K sample high-quality dataset.
- We fine-tune LLaVA-V1.5, Qwen2VL, and Qwen2.5VL on VAPR and evaluate them across ten benchmarks. VAPR outperforms SFT and preference-tuned baselines on 8 out of 10 benchmarks, with average gains of 6.5% (LLaVA), 4.0% (Qwen2VL), and 1.5% (Qwen2.5VL). Scaling analysis shows LLaVA benefits from smaller-scale data, while Qwen models require larger-scale data to improve.
- Our analysis shows that VAPR enhances performance on reasoning tasks like adversarial, spatial, counting, & textual, reduces the overuse of "Yes" in binary questions, and avoids spurious learning signals by mitigating stylistic and length biases present in prior preference datasets.
- Our ablation with Qwen3-32b as the LLM-editor, demonstrates that the framework generalizes well to open-source LLMs, producing similarly effective hard negatives (VAPR-OS) and preference-tuned models achieving 99% of the performance of models trained on VAPR. Our data, models, and code can be found on the project page https://vap-r.github.io/

2 Preliminaries

Large Vision Language Models (LVLMs) enhance the capabilities of Large Language Models (LLMs) by adapting them to multimodal tasks. This enables the model to predict the probability distribution for the next token in a sequence given multimodal inputs. Specifically, given an input pair $x \langle x_v, x_t \rangle$ comprising an image x_v and instruction text x_t, the LVLM generates a text response y. Preference optimization has emerged as a promising technique for fine-tuning language models to align their outputs with desired outcomes, which we briefly overview in this section.

2.1 Direct Preference Optimization

Given a prompt $x \langle x_v, x_t \rangle$, a large vision language model governed by policy π_θ can yield a conditional distribution $\pi_\theta(y \mid x)$, where y is the generated text response. To train using

preference data, we define the dataset $\mathcal{D} = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^{N}$, where $y_w^{(i)}$ and $y_l^{(i)}$ represent the more and less preferred responses, in our case, chosen and hard-negative rejected response, respectively, for a given input $x^{(i)}$. Preference optimization leverages this dataset to fine-tune models effectively.

Objective Direct Preference Optimization (DPO) calculates the probability of preferring y_w over y_l as:

$$p(y_w \succ y_l) = \sigma(r(x, y_w) - r(x, y_l)),$$

where $\sigma(\cdot)$ is the sigmoid function. The loss function for DPO can be written as follows:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma \left(\alpha \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \alpha \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right], \quad (1)$$

where $\pi_{\text{ref}}(y \mid x)$ represents the reference policy, typically the model after SFT.

Preference Dataset There are different ways to construct the preference dataset. For example, Sun et al. (2023) employs human annotations, which are of high quality but is difficult to scale. On the other hand, AI annotation with LVLMs like GPT-4V (OpenAI, 2023) scales better but risks inconsistencies and hallucinations (Zhao et al., 2023; Hong et al., 2024). Our focus is to construct a scalable framework for preference dataset construction while ensuring its quality.

3 VaPR Dataset Construction

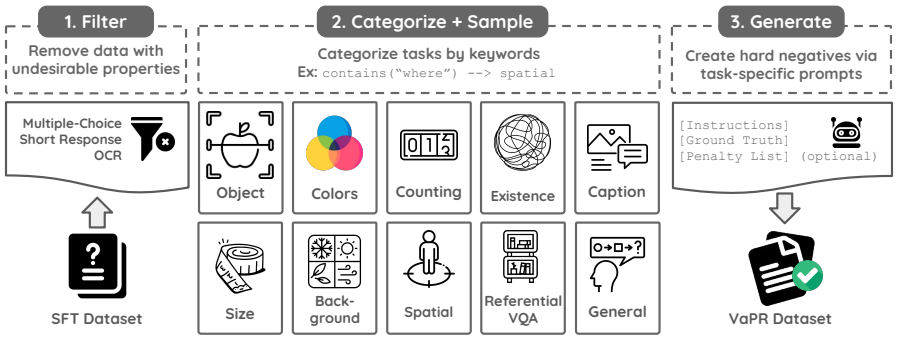


Figure 2: VaPR: A three-stage pipeline that generates 30K hard-negative preference pairs from LLaVA-v1.5-665K SFT (Liu et al., 2023b). **Stage 1:** Filter out irrelevant samples (e.g., MCQs). **Stage 2:** Categorize remaining samples based on task. **Stage 3:** Use task-specific prompts (with optional penalty lists) to produce stylistically and length-wise similar but content-distinct negative responses.

We detail the design principles and construction processes of our preference dataset. Our main idea is to construct hard negatives that are stylistically and length-wise similar to the ground-truth samples.

3.1 Design Principles

We adhered to two key principles for constructing the VaPR dataset:

Task Diversity The dataset ensures a near-balanced distribution across a wide range of tasks (see Fig. 3). It consists of foundational vision-language capabilities, including perceptual (e.g., object recognition, object attribute recognition - color, size, etc., and

background understanding) and reasoning (e.g., spatial, counting, comparative). We also incorporate tasks combining perception and reasoning like captioning, world knowledge reasoning, and referential VQA (eg. region-level perception or reasoning). This diversity is designed to improve the model's comprehensive capabilities.

Hard Negative Rejected Responses Inspired by the role of hard negatives in enhancing vision-language compositionality in image-text pretraining (Radford et al., 2021; Yuksekgonul et al., 2022; Hsieh et al., 2023), we generate hard-negative responses. These are synthesized by introducing targeted perturbations to high-quality SFT ground truth responses while maintaining stylistic and structural similarity (e.g., length), thereby resulting in incorrect responses given a task.

3.2 Sourcing & Processing

Sourcing We build preference samples using the high-quality LLaVA-665K SFT dataset (Liu et al., 2023b), which has broad task coverage.

Filtering The data undergoes careful processing to filter tasks unsuitable for hard-negative generation. We exclude tasks that lack sufficient training signals to address vision-language misalignments, such as text-only tasks and simple response types (e.g., MCQ or bounding box prediction). However, we retain "Yes/No" response instruction on the existence of objects, attributes, and reasoning (e.g., count, spatial, comparative), as they correspond to the key vision-language capabilities, and convert them to extended responses to better suit preference optimization. We also filter out OCR instructions, as prior works emphasize the need for larger input image resolutions and fine-grained visual perception as driving factors in improving OCR performance (Wadhawan et al., 2024; Li et al., 2024b; Yu et al., 2024c).

Categorization We categorize the filtered corpus into ten task categories using task-specific keywords (see Fig. 2). We subsample a portion of the SFT dataset from each category for hard-negative response generation. Notably, for binary response instructions ("Yes"/"No"), we enforce an equal distribution of "Yes" and "No" responses to mitigate bias towards "Yes," which arises from the predominance of affirmative instructions in the original SFT dataset (Li et al., 2023d; Guan et al., 2024; Liu et al., 2023a). See Appendix §B.1 for details.

3.3 Generation Pipeline

The VAPR framework generates hard-negative responses by editing ground truth (instruction, response) pairs from our filtered & categorized SFT subset using GPT-4o. This process is guided by two key components: conditioning information and perturbation diversity.

Conditioning Information Task-specific prompts (e.g., attributes, spatial reasoning, counting) guide edits to ensure responses remain fluent yet incorrect (see Fig. 2). These were preferred over general prompts, which often introduced irrelevant changes given the task or added verbosity or style changes, contributing to noise in the dataset.

Diversity of Perturbations Most categories (e.g., object type, size, existence) produced diverse outputs with zero-shot prompting, while for others (e.g., color, counting, captioning), we introduced a penalty list, periodically updating it with commonly used perturbation values to prevent reuse. In captioning, we first extracted dimensions (e.g., color, spatial relation) and applied random perturbations across them, using a penalty list to encourage a variety of dimensions and perturbation values. See Appendix §B.2 for details.

3.4 Dataset Analysis

We perform statistical analysis of the VAPR dataset, with task-wise distribution shown in Fig. 3. To assess annotation quality, we conducted a human evaluation on 500 stratified samples across task categories, finding 97% alignment with hard-negative criteria and 86% inter-annotator agreement (IAA) using Fleiss' kappa (see Appendix §B.4).

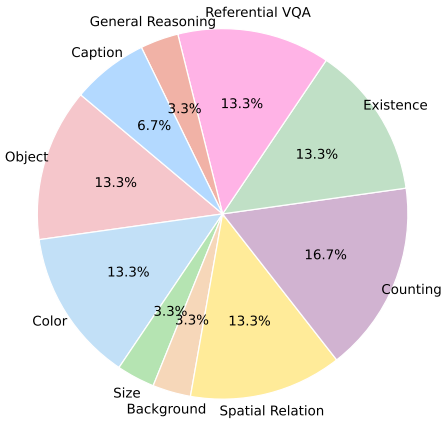


Figure 3: VAPR task distribution

Table 1: Comparison of the VAPR Preference Dataset with Related Works. Lower LD and smaller length differences indicate higher stylistic similarity. We show the percentage of samples where chosen responses are longer (*chosen* > *rejected*) or shorter (*rejected* > *chosen*).

	Ours	HA-DPO	POVID	RLAIF-V	CSR
Overall Samples (%)	(21, 79)	(41, 59)	(24, 76)	(45, 55)	(38, 62)
- Linguistic Similarity	6	49	30	62	97
- Avg. Token Length Difference	3	18	16	15	27
- Token Length Difference	(10, 1)	(15, 20)	(27, 13)	(17, 14)	(23, 29)

To quantify stylistic and length similarity, we compare VAPR against prior datasets - HA-DPO (Zhao et al., 2023), POVID (Zhou et al., 2024a), RLAIF-V (Yu et al., 2024b), and CSR (Zhou et al., 2024b). Linguistic similarity is measured via word-level Levenshtein distance, where lower values indicate targeted content edits; higher values reflect broader stylistic variation. Length similarity is computed as token-level sequence length differences. We also report the proportion of samples where chosen or rejected responses are longer, for both short-form (e.g., VQA) and long-form (e.g., captioning) tasks (see Appendix §B.3). Table 1 shows that VAPR has the lowest Levenshtein distance and token length difference, reinforcing the hard-negative nature of its rejections. In Section §4.3.1, we show that larger stylistic and length discrepancies in other datasets lead to premature saturation in reward accuracy, suggesting reward hacking behavior that is mitigated with VAPR.

4 Experiments

4.1 Experimental Setup

4.1.1 Models & Baselines

We preference-tune LLaVA-1.5 (Liu et al., 2023b) (7B and 13B), Qwen2VL-Instruct (Wang et al., 2024a) (2B and 7B), and Qwen2.5VL-Instruct (3B and 7B) using Direct Preference Optimization (DPO) (Rafailov et al., 2024). LoRA finetuning (Hu et al., 2021) is employed during the preference learning phase. Models preference-tuned on the VAPR dataset, which comprises 30K samples, are referred to as LLaVA-VAPR, Qwen2VL-VAPR, and Qwen2.5VL-VAPR, respectively. Additional details on the training setup are provided in Appendix §C.1.

We compare the performance of the VAPR models with two baselines: base instruct models and Supervised Finetuned (SFT) variants of base instruct models on the VAPR dataset. We also compare the VAPR models with prior works involving different preference dataset generation techniques - Human (LLaVA-RLHF (Sun et al., 2023)), AI annotations using a large closed-source model (HA-DPO (Zhao et al., 2023), POVID (Zhou et al., 2024a)), self-generated AI feedback (SIMA (Wang et al., 2024b), RLAIF-V (Yu et al., 2024b), and CSR (Zhou et al., 2024b)) - using models made publicly available by the respective works.

4.1.2 Evaluation Benchmarks

We evaluate VAPR models across ten benchmarks covering diverse skills: Open-ended QA - LLaVA-in-the-wild (LLaVA^{W}) (Liu et al., 2023c), ConTextual (ConT) (Wadhawan et al., 2024), & MM-VET (MMV) (Yu et al., 2023); Comprehensive vision benchmarks (reasoning & perception) - SEED Bench (SEED^{I} - image split) (Li et al., 2023a) & MMStar (MMS) (Chen

Table 2: Performance comparison of LLaVA-v1.5-Instruct, Qwen2VL-Instruct, Qwen2.5VL-Instruct, SFT and DPO models finetuned on VAPR, and other preference datasets across 2B, 3B, 7B, and 13B parameter sizes on ten benchmarks. Higher scores indicate better performance across all benchmarks, with the highest score for each benchmark highlighted in **bold**. All models are evaluated using publicly available checkpoints, adhering to evaluation parameters prescribed by the benchmarks. † represents results that show statistically significant improvement via bootstrap resampling (95% CI).

Row	Method	LLaVA^{W}	ConT	MMV	SEED^{I}	CV	MV	MMMU	MMS	POPE	NB
1	LLaVA-1.5-7B	64.8	16.8	30.9	66.2	62.1	30.1	35.4	32.6	85.9	12.7
2	+ Fact-RLHF	58.7	10.5	30.2	55.7	35.1	26.4	28.0	30.7	79.8	7.8
3	+ HA-DPO	69.8	14.2	30.9	64.6	62.0	29.7	35.5	32.9	83.5	13.5
4	+ POVID	70.1	18.6	31.8	66.3	62.3	30.6	35.6	33.3	86.0	13.1
5	+ SIMA	66.4	18.0	31.4	66.1	60.1	30.1	35.6	32.5	85.8	12.6
6	+ RLAIF-V	73.1	14.1	29.3	65.8	59.8	30.3	34.2	33.7	78.9	12.9
7	+ CSR	69.5	15.8	30.6	65.9	61.1	30.5	35.6	32.2	**86.2**	13.4
8	+ VAPR SFT (ours)	64.0	16.0	30.0	65.6	59.5	29.5	34.9	31.7	83.1	12.2
9	+ VAPR DPO (ours)	**76.2**†	**20.6**†	**32.9**	**66.7**†	**62.9**†	**30.8**	**35.7**	**34.7**†	85.4	**14.5**†
10	LLaVA-1.5-13B	72.3	18.6	36.7	68.2	62.5	30.7	**36.1**	33.8	86.0	14.9
11	+ Fact-RLHF	70.4	15.6	37.0	61.1	53.7	31.2	28.0	32.1	81.7	12.4
12	+ SIMA	70.7	19.2	36.0	68.0	62.6	30.8	35.4	34.0	86.0	14.6
13	+ CSR	74.2	17.8	35.6	68.2	62.4	31.3	35.9	33.9	**86.8**†	14.9
14	+ VAPR SFT (ours)	71.4	17.4	34.7	67.2	61.5	30.4	34.5	33.3	83.9	13.7
15	+ VAPR DPO (ours)	**80.5**†	**21.2**	**37.3**	**68.7**†	**64.6**†	**32.3**†	35.8	**35.6**†	86.3	**18.2**†
16	Qwen2VL-2B	83.2	27.7	53.3	73.6	66.5	**51.0**	38.7	43.4	86.5	24.3
17	+ VAPR SFT (ours)	69.1	22.1	43.3	70.3	54.5	46.4	36.6	41.5	85.5	15.1
18	+ VAPR DPO (ours)	**88.1**	**34.8**†	**54.1**	**74.0**†	**69.0**†	50.9	**39.2**	**43.7**	**88.3**†	**25.7**†
19	Qwen2VL-7B	92.5	39.7	62.1	76.4	75.7	57.5	**50.7**	56.7	**87.3**	30.8
20	+ VAPR SFT (ours)	79.6	36.6	52.6	73.9	67.4	56.3	47.7	52.9	86.2	23.7
21	+ VAPR DPO (ours)	**96.2**	**43.9**†	**65.4**	**76.8**†	**76.3**†	**58.2**	50.0	**57.8**†	**87.3**	**32.5**†
22	Qwen2.5VL-3B	98.1	37.2	67.3	75.0	71.5	52.5	**45.7**	54.7	86.3	25.4
23	+ VAPR SFT (ours)	79.8	33.4	50.1	70.9	60.5	48.6	43.1	48.5	84.8	21.2
24	+ VAPR DPO (ours)	97.1	**40.3**	**67.4**	**75.5**†	**72.7**†	**53.4**	44.9	**56.1**†	**86.4**	**26.3**†
25	Qwen2.5VL-7B	101.4	53.3	71.0	77.7	80.1	58.6	**50.9**	61.9	86.3	32.0
26	+ VAPR SFT (ours)	80.2	38.9	59.0	74.9	69.4	56.8	48.1	56.1	82.8	23.8
27	+ VAPR DPO (ours)	**101.5**	**53.4**	**72.4**	**77.8**	**81.1**†	**59.8**†	50.6	**62.5**	**86.9**†	**32.8**†

et al., 2024a); , Vision-centric reasoning (spatial reasoning, counting) - CV Bench (CV) (Tong et al., 2024); Hallucination & Adversarial reasoning - Pope (Li et al., 2023d), NaturalBench (NB) (Li et al., 2024a); Academic & Math Reasoning - MathVista (MV) (Lu et al., 2023) & MMMU (Yue et al., 2024) (benchmark details provided in Appendix §C.2).

4.2 Results

Table 2 presents the core experimental findings. VAPR models outperform baseline and prior preference-tuned models on 8 out of 10 benchmarks and remain competitive on the rest. LLaVA-VAPR 7B and 13B achieve average gains of 7% and 6%, respectively, while Qwen2VL-VAPR 2B and 7B improve by 5% and 3%. Despite Qwen2.5VL being a strong baseline, preference finetuning on VAPR yields improvements of 2% (3B) and 1% (7B), with gains concentrated in vision-centric benchmarks (CV-Bench, MMStar, SeedBench), and the adversarial benchmark (NaturalBench).

These improvements in vision-centric and adversarial benchmarks are consistent across all VAPR models, indicating that preference finetuning on VAPR strengthens visio-linguistic compositionality - particularly in perception (e.g., fine-grained object-attribute identification) and reasoning tasks such as spatial relationships and counting (see Appendix §D.2). Notably, improvements also extend to textual and mathematical reasoning benchmarks (ConTextual and MathVista), despite not being explicitly trained on OCR, textual reasoning, or math tasks, VaPR models achieve strong performance in these areas. We attribute this to their enhanced fine-grained perception, spatial reasoning, and counting capabilities, which is consistent with prior work (Fu et al., 2024). In contrast, all models show limited or no improvement on MMMU, aligning with prior findings (Ivison et al., 2024a) that preference optimization primarily enhances alignment and truthfulness, rather than factuality.

Lastly, we observe that While VAPR demonstrates strong effectiveness under DPO optimization, supervised finetuning (SFT) on the same dataset does not yield comparable gains. Specifically, SFT results in gentle performance degradation for LLaVA, and more for the Qwen2VL and Qwen2.5VL families. This outcome suggests that SFT on a relatively small dataset like VAPR- particularly in comparison to the original large-scale SFT corpora - may lead to overfitting, especially in models with strong pretrained priors like Qwen2.5VL, thereby limiting their generalization capabilities. In contrast, preference optimization with carefully constructed, length and stylistically similar hard-negative pairs enables more robust and generalizable representation learning. These findings indicate that the gains observed in VAPR models stem not from the selection of samples, but from the preference signal induced by VAPR preference pairs.

4.3 Analysis

4.3.1 Preference Dataset Comparison

We compare VAPR with two alternatives: (1) **POVID**, which uses GPT-4V to generate rejected responses given an SFT sample, and (2) **SIMA**, a self-preference method that composes preference pairs from greedy and sampled outputs generated and critiqued by the same LVLM to be preference tuned. VAPR consistently outperforms both LLaVA and Qwen2VL families, showing 4-6% average gains over POVID and SIMA.

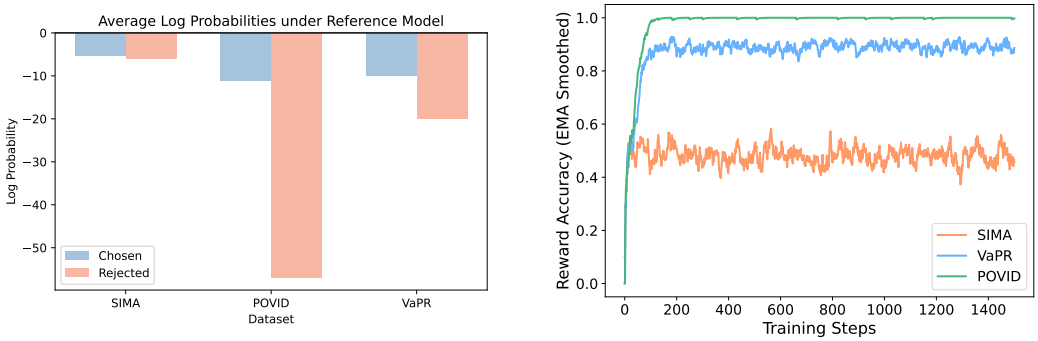
To analyze how the different datasets affect the DPO optimization process, let us rewrite the loss:$\mathcal{L}_{\text{DPO}} = -\log \sigma\left(\alpha(\Delta_\theta - \Delta_{\text{ref}})\right)$, where $\Delta_\theta = \log \pi_\theta(y_w \mid x) - \log \pi_\theta(y_l \mid x)$ and $\Delta_{\text{ref}} = \log \pi_{\text{ref}}(y_w \mid x) - \log \pi_{\text{ref}}(y_l \mid x)$. We observe that POVID exhibits higher Δ_{ref} than VAPR despite similar chosen log-probabilities (see Fig. 4a), indicating that its rejected responses are less likely under the reference model most likely due to greater stylistic and length differences (see Table 1). We further observe that the POVID model rapidly attains a reward accuracy of 1 (see Fig. 4b), suggesting that the model may be overfitting based on preference signals derived from length and stylistic differences, unlike VAPR models, which improve more gradually without saturating, indicating reduced overfitting due to exposure to more challenging preference pairs.

SIMA, on the other hand, shows $\Delta_{ref} \approx 0$ (see Fig. 4a), with chosen and rejected responses often near-identical - including 20% duplicates. In such cases, the loss depends entirely on Δ_θ, removing reference-guided regularization and amplifying noisy or weak signals. This leads to poor reward accuracy (50%) and degraded generalization (see Fig. 4). While self-preference methods like CSR and RLAIF-V can mitigate some issues via multi-step scoring, like POVID, they can still reward hack due to length and stylistic differences (see Table 1). Detailed methodology, results & analysis is provided in Appendix §D.3.

VAPR addresses both issues by explicitly generating hard negatives that are stylistically and length-wise similar to positives, ensuring DPO learns from content differences. In low-resource settings where high-quality SFT data is scarce, VAPR can complement self-preference methods by using confident responses from approaches like CSR or RLAIF-V as positives and generating hard negatives via the VaPR pipeline, allowing the curation of high-quality preference data in an unsupervised manner - a promising future direction

4.3.2 Scaling analysis

We investigate the impact of dataset size on model performance by conducting a data scaling analysis using three training budgets: 3K, 10K, and 30K samples from the VAPR dataset. Detailed results are provided in Appendix §D.1. As shown in Fig. 5, all VAPR models exhibit improved performance with increasing data. Interestingly, LLaVA-VAPR models achieve substantial gains even at the lowest data budget (3K), with diminishing returns at higher scales. In contrast, Qwen2VL and Qwen2.5VL models- being stronger base instruct models- show more modest improvements at 3K but benefit more as dataset size increases. This trend aligns with their stronger pretrained priors compared to LLaVA-v1.5, which may require more supervision to meaningfully shift under preference tuning.



(a) Reference model log-probabilities

(b) Reward accuracy trends

Figure 4: Comparison of preference datasets. (a) Average reference model log-probabilities for chosen vs. rejected responses across VaPR, SIMA, and POVID - lower values indicate lower reference likelihood. (b) Reward accuracy trends over training steps show that SIMA improves gradually while POVID saturates quickly.

4.3.3 *VaPR models do not overconfidently say "Yes" to "Yes/No" questions*

In this section, we examine the response patterns of large vision-language models (LVLMs), which tend to answer "Yes" more frequently than "No" for binary "Yes/No" questions (Li et al., 2023d; 2024a; Liu et al., 2023a; Guan et al., 2024). Specifically, we analyze model outputs on NaturalBench, an adversarial benchmark comprising paired instructions and paired images. As shown in Fig. 6, base SFT variants of LLaVA-v1.5, Qwen2VL, and Qwen2.5VL exhibit a clear tendency to favor "Yes" responses, even for questions where the correct answer is "No" (highlighted in red). However, the VaPR models demonstrate a notable reduction in this bias, with an emergent shift towards answering "No" more frequently than "Yes." This behavior is most pronounced in LLaVA-VaPR 13B. We attribute these improvements to enhanced perception and reasoning capabilities, which manifest as improved visio-linguistic compositionality and a reduced bias towards "Yes" responses.

4.3.4 *VaPR-OS: Ablation Study using open-source LLM Editor*

We conduct an ablation using Qwen3-32B as the open-source LLM editor to evaluate the generalizability of the VaPR pipeline beyond closed models like GPT-4o (see Appendix §D.4). The resulting dataset, VaPR-OS, is constructed from the same subset as the 10K version of VaPR. Analyzing the data, we find that VaPR-OS exhibits comparable hard-negative characteristics to VaPR, with a token length gap of 6 (vs. 3 in VaPR) and a Levenshtein distance of 10 (vs. 6). We fine-tuned LLaVA-v1.5-Instruct-7B, Qwen2VL-Instruct-2B, and Qwen2.5VL-Instruct-3B on VaPR-OS and evaluated their performance against models trained on the GPT-4o-based VaPR (10K subset) across benchmarks (§D.4). Results show that models trained on VaPR-OS achieve 99% of the performance of their GPT-4o-based counterparts (VaPR), with both consistently outperforming base instruct models. These findings demonstrate that the VaPR pipeline generalizes well to open-source editors, allowing researchers to apply the VaPR framework to their own SFT datasets without relying on closed-source APIs.

5 Related Work

Large Vision Language Models (LVLMs) LVLMs combine visual inputs from pre-trained vision encoders (Radford et al., 2021; Zhai et al., 2023) with large language models (LLMs) (Team, 2024b; Touvron et al., 2023; Dubey et al., 2024; Cai et al., 2024) via projection modules (Liu et al., 2023b; Ye et al., 2024; Li et al., 2023b; Tong et al., 2024). They are typically trained in two stages: pretraining on large-scale image-text data to align modalities and instruction tuning or supervised finetuning on vision-language datasets for open-ended tasks.

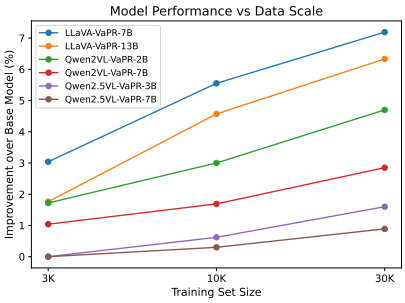


Figure 5: Performance scaling of VAPR models with 3K, 10K, and 30K samples, shown as % improvement over base instruct models. Note: X-axis spacing between 3K, 10K, and 30K is not uniform.

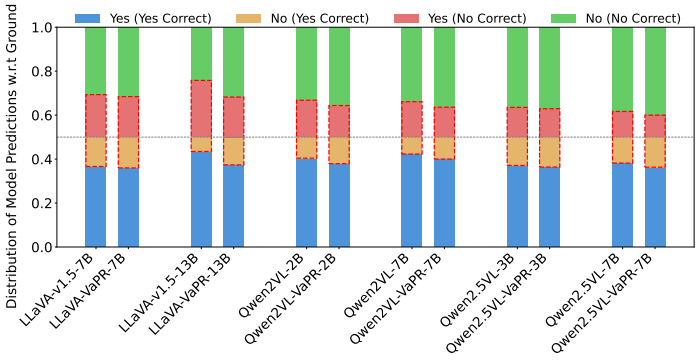


Figure 6: Comparison of "Yes"/"No" responses on Natural-Bench - base vs. VAPR models. Dotted rectangles highlight incorrect "Yes" (red) and "No" (yellow) predictions, illustrating reduced "Yes" bias after preference finetuning.

Preference Optimization in LVLMs Preference optimization, typically the third training stage for LVLMs, improves modality alignment and reduces hallucinations. It relies on reference datasets sourced via human annotation (Yu et al., 2024a; Sun et al., 2023), AI annotation (Zhou et al., 2024a; Li et al., 2023c; Zhao et al., 2023), or self-curation (Yu et al., 2024b; Zhou et al., 2024b; Deng et al., 2024; Wang et al., 2024b). While human labels are high-quality but expensive, AI annotations using models like GPT-4V (OpenAI, 2023) are scalable but may introduce stylistic or length inconsistencies and distill hallucinations. Self-curated approaches generate multiple responses, rank them heuristically or with LVLMs, and select preference pairs, but often fail to control for stylistic or length biases - leading to noise, as both chosen and rejected responses may seem equally plausible (Yan et al., 2024). Our method builds on Direct Preference Optimization (DPO) (Rafailov et al., 2024) and introduces a post-SFT data construction pipeline that addresses these challenges. Unlike prior works (Zhou et al., 2024a; Yu et al., 2024b; Zhou et al., 2024b; Deng et al., 2024; Wang et al., 2024b) that use VLMs for generation or scoring, we employ LLM-guided response editing to inject task-aware semantic errors into rejected responses while explicitly preserving style and length. This reduces the risk of DPO exploiting superficial cues. In contrast to Chen et al. (2025), which uses LLMs to generate hard negatives for multilingual embedding training, our approach targets post-SFT preference alignment in English, where fine-grained semantic and stylistic control is crucial.

6 Conclusion

In this work, we address challenges in the alignment and reasoning capabilities of LVLMs by introducing VAPR, a hard-negative preference data generation framework. By constructing a high-quality preference dataset of 30K samples, we mitigate length and stylistic biases prevalent in existing synthetic datasets. We demonstrate significant improvements across benchmarks, particularly excelling in reasoning tasks and adversarial scenarios. Our analysis highlights the effectiveness of VAPR in improving vision-linguistic compositionality and reducing binary question bias, paving the way for more reliable and generalizable LVLMs.

Future work includes extending the framework to larger and more diverse datasets, applying it to broader reasoning tasks, and refining hard-negative generation to better target nuanced reasoning errors. Another promising direction is combining VAPR with self-preference methods in low-resource settings - using confident responses as positives and generating hard negatives via the VAPR pipeline to enable unsupervised preference data creation. We also plan to explore integrating VAPR with online preference optimization methods such as PPO and GRPO, particularly for complex reasoning tasks like math.

Acknowledgments

This research is supported in part by the ECOLE program under Cooperative Agreement HR00112390060 with the US Defense Advanced Research Projects Agency (DARPA), UCLA-Amazon Science Hub for Humanity and Artificial Intelligence, and OpenAI Researcher Access Program (Grant No. 0000004868).

References

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, Chao Xu, Ruiliang Xu, Hang Yan, Yirong Yan, Xiaogui Yang, Haochen Ye, Huaiyuan Ying, Jia Yu, Jing Yu, Yuhang Zang, Chuyu Zhang, Li Zhang, Pan Zhang, Peng Zhang, Ruijie Zhang, Shuo Zhang, Songyang Zhang, Wenjian Zhang, Wenwei Zhang, Xingcheng Zhang, Xinyue Zhang, Hui Zhao, Qian Zhao, Xiaomeng Zhao, Fengzhe Zhou, Zaida Zhou, Jingming Zhuo, Yicheng Zou, Xipeng Qiu, Yu Qiao, and Dahua Lin. Internlm2 technical report, 2024. URL https://arxiv.org/abs/2403.17297.

Haonan Chen, Liang Wang, Nan Yang, Yutao Zhu, Ziliang Zhao, Furu Wei, and Zhicheng Dou. mme5: Improving multimodal multilingual embeddings via high-quality synthetic data. *arXiv preprint arXiv:2502.08468*, 2025.

Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024a.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites, 2024b. URL https://arxiv.org/abs/2404.16821.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. URL https://arxiv.org/abs/2305.06500.

Yihe Deng, Pan Lu, Fan Yin, Ziniu Hu, Sheng Shen, James Zou, Kai-Wei Chang, and Wei Wang. Enhancing large vision language models with self-training on image comprehension. *arXiv preprint arXiv:2405.19716*, 2024.

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

Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, et al. Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. *arXiv preprint arXiv:2501.00321*, 2024.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models, 2024. URL https://arxiv.org/abs/2310.14566.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model, 2024. URL https://arxiv.org/abs/2403.07691.

Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality, 2023. URL https://arxiv.org/abs/2306.14610.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A Smith, Yejin Choi, and Hanna Hajishirzi. Unpacking dpo and ppo: Disentangling best practices for learning from preference feedback. *Advances in neural information processing systems*, 37:36602–36633, 2024a.

Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A Smith, Yejin Choi, and Hannaneh Hajishirzi. Unpacking dpo and ppo: Disentangling best practices for learning from preference feedback. *arXiv preprint arXiv:2406.09279*, 2024b.

Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 388–395, 2004.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv preprint arXiv:2406.18629*, 2024.

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

Seongyun Lee, Sue Hyun Park, Yongrae Jo, and Minjoon Seo. Volcano: mitigating multimodal hallucination through self-feedback guided revision. *arXiv preprint arXiv:2311.07362*, 2023b.

Baiqi Li, Zhiqiu Lin, Wenxuan Peng, Jean de Dieu Nyandwi, Daniel Jiang, Zixian Ma, Simran Khanuja, Ranjay Krishna, Graham Neubig, and Deva Ramanan. Naturalbench: Evaluating vision-language models on natural adversarial samples. *arXiv preprint arXiv:2410.14669*, 2024a.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023a.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023b.

Lei Li, Zhihui Xie, Mukai Li, Shunian Chen, Peiyi Wang, Liang Chen, Yazheng Yang, Benyou Wang, and Lingpeng Kong. Silkie: Preference distillation for large visual language models. *arXiv preprint arXiv:2312.10665*, 2023c.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models, 2023d. URL https://arxiv.org/abs/2305.10355.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26763–26773, 2024b.

Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Mitigating hallucination in large multi-modal models via robust instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2023a.

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

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023c.

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/.

Junru Lu, Jiazheng Li, Siyu An, Meng Zhao, Yulan He, Di Yin, and Xing Sun. Eliminating biased length reliance of direct preference optimization via down-sampled kl divergence. *arXiv preprint arXiv:2406.10957*, 2024.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

OpenAI. Gpt-4v(ision) system card, 2023b. https://openai.com/research/gpt-4v-system-card, 2023.

OpenAI. Gpt-4o system card. https://openai.com/index/gpt-4o-system-card/, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization, 2024. *URL https://arxiv. org/abs/2403.19159*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Pritam Sarkar, Sayna Ebrahimi, Ali Etemad, Ahmad Beirami, Sercan Ö. Arık, and Tomas Pfister. Mitigating object hallucination via data augmented contrastive tuning, 2024. URL https://arxiv.org/abs/2405.18654.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, et al. Aligning large multimodal models with factually augmented rlhf. *arXiv preprint arXiv:2309.14525*, 2023.

OpenGVLab Team. Internvl2: Better than the best—expanding performance boundaries of open-source multimodal models with the progressive scaling strategy. https://internvl.github.io/blog/2024-07-02-InternVL-2.0/, 2024a.

The Vicuna Team. Vicuna: An open-source chatbot impressing gpt-4 with 90 https://lmsys.org/blog/2023-03-30-vicuna/, 2024b.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Rohan Wadhawan, Hritik Bansal, Kai-Wei Chang, and Nanyun Peng. Contextual: Evaluating context-sensitive text-rich visual reasoning in large multimodal models. *arXiv preprint arXiv:2401.13311*, 2024.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

Xiyao Wang, Jiuhai Chen, Zhaoyang Wang, Yuhang Zhou, Yiyang Zhou, Huaxiu Yao, Tianyi Zhou, Tom Goldstein, Parminder Bhatia, Furong Huang, et al. Enhancing visual-language modality alignment in large vision language models via self-improvement. *arXiv preprint arXiv:2405.15973*, 2024b.

Siming Yan, Min Bai, Weifeng Chen, Xiong Zhou, Qixing Huang, and Li Erran Li. Vigor: Improving visual grounding of large vision language models with fine-grained reward modeling, 2024. URL https://arxiv.org/abs/2402.06118.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, and Fei Huang. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13040–13051, 2024.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, et al. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13807–13816, 2024a.

Tianyu Yu, Haoye Zhang, Yuan Yao, Yunkai Dang, Da Chen, Xiaoman Lu, Ganqu Cui, Taiwen He, Zhiyuan Liu, Tat-Seng Chua, et al. Rlaif-v: Aligning mllms through open-source ai feedback for super gpt-4v trustworthiness. *arXiv preprint arXiv:2405.17220*, 2024b.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

Ya-Qi Yu, Minghui Liao, Jiwen Zhang, and Jihao Wu. Texthawk2: A large vision-language model excels in bilingual ocr and grounding with 16x fewer tokens. *arXiv preprint arXiv:2410.05261*, 2024c.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9556–9567, 2024.

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? *arXiv preprint arXiv:2210.01936*, 2022.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023.

Zhiyuan Zhao, Bin Wang, Linke Ouyang, Xiaoyi Dong, Jiaqi Wang, and Conghui He. Beyond hallucinations: Enhancing lvlms through hallucination-aware direct preference optimization. *arXiv preprint arXiv:2311.16839*, 2023.

Yiyang Zhou, Chenhang Cui, Rafael Rafailov, Chelsea Finn, and Huaxiu Yao. Aligning modalities in vision large language models via preference fine-tuning. *arXiv preprint arXiv:2402.11411*, 2024a.

Yiyang Zhou, Zhiyuan Fan, Dongjie Cheng, Sihan Yang, Zhaorun Chen, Chenhang Cui, Xiyao Wang, Yun Li, Linjun Zhang, and Huaxiu Yao. Calibrated self-rewarding vision language models. *arXiv preprint arXiv:2405.14622*, 2024b.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

A Appendix Overview

The appendix is organized into the following sections and subsections:

- **Detailed Methodology of VAPR Data Generation** (§B):
  - **Task Categories (§B.1)**: Definition of the task categories included in the VAPR dataset.
  - **Prompt Design & Examples (§B.2)**: Specific prompts used for data generation for each task category and corresponding qualitative examples illustrating the generated data.
  - **Fine-grained length & Stylistic analysis (§B.3)**
  - **Human Study (§B.4)**: Annotation and evaluation setup.
- **Experimental Setup** (§C):
  - **Training Setup (§C.1)**: Details of the training setup for our models are provided here.
  - **Evaluation Setup (§C.2)**: Implementation of the evaluation benchmarks.
- **Extended Results** (§D):
  - **Data Scaling (§D.1)**: Detailed results for VAPR model with increasing training dataset size.
  - **Benchmark detailed results (§D.2)**: Fine-grained evaluation across benchmark categories, for CVBench and MMStar.
  - **Preference Dataset Comparison (§D.3)**: Detailed results for the comparison of base models trained on VAPR and other preference datasets.

B Detailed Methodology of VAPR Data Generation

In this section, we provide the breakdown of VAPR dataset into task categories, prompt design & examples for each category, setup for human study and linguistic & length analysis study.

B.1 Task Categories

An essential component of our data generation pipeline is the categorization of filtered samples (~270K) from the LLaVA-665K SFT set into distinct task categories. This ensures comprehensive coverage of perception, reasoning, and composite tasks (perception & reasoning) in the final VAPR dataset. The VAPR dataset comprises ten task categories: object (like type, material, action), color, size, background (like weather, time of day, surrounding lighting), counting, spatial reasoning, existence, referential VQA (like color of an object on the left, etc), general reasoning (like abstract & knowledge-based), and image captioning, as outlined in Table 3. The categorization process is carried out in the following three steps:

- Task-specific keywords (see Table 3) are applied to the instructions in the filtered SFT set to assign a task category to each sample containing those keywords.
- Categorization follows a defined order:
  - Samples are initially tagged into the categories: color, size, counting, spatial reasoning, background, existence, captioning, referential VQA (comparative reasoning) and general reasoning (in no specific order). Specifically for the existence category, we check the first word in the instruction.
  - Remaining samples are assigned to the object category. This approach is necessary because object-related tasks, encompassing types and knowledge, require a broad, non-generalizable keyword list.
- **Resolving Multiple Task Categories for a sample**: Samples tagged with multiple task categories are categorized as referential VQA (qualitative examples shown in §B.2.8). Thus, referential VQA consists of samples tagged as comparative reasoning and the above samples.

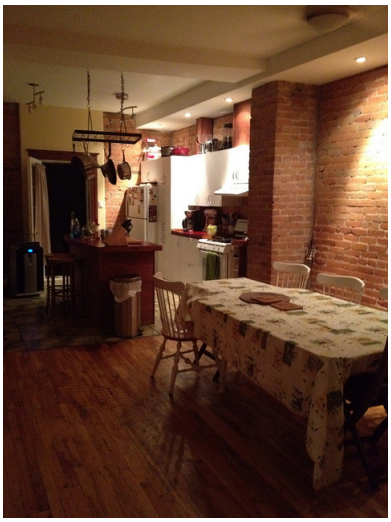
Task	**Task Type**	**Keyword**
Color	Perception	color(s)
Size	Perception	size(s)
Background	Perception	environment, time of, day, year, weather, lighting,
Counting	Reasoning	many, count(s), instance(s), counting
Spatial Reasoning	Reasoning	where, located, placed, positioned, left, right, in front of, down, above, below
Existence	Perception & Reasoning	are, is, can, do, does, would, will
General Reasoning	Perception & Reasoning	could, would, might, purpose, reason, based, should
Referential VQA	Perception & Reasoning	Samples with keywords from more than one task Comparative Reasoning: comparison, difference, closer, nearer, bigger
Image Captioning	Perception & Reasoning	analyze, describe, write, elaborate, description, snapshot

Table 3: Overview of tasks, task types (perception & reasoning), and task-specific keywords used for VAPR dataset categorization. Keyword list is shortened for clarity. Note untagged samples are grouped together into the object category.

B.2 Prompt Design & Examples

In this section, we provide a detailed walkthrough of hard-negative generation for each task category, accompanied by additional examples from each category - object (§B.2.1), color (§B.2.2), size (§B.2.3), background (§B.2.4), counting (§B.2.5), spatial reasoning (§B.2.6), existence (§B.2.7), referential VQA (§B.2.8), image captioning (§B.2.9), general reasoning (§B.2.10). In the prompt figures, spans of the chosen responses are highlighted in blue, while the corresponding perturbed spans in the generated hard-negative responses are highlighted in green. Lastly, we also provide the algorithm and additional examples showing generation steps for task categories with penalty list (e.g. color) in §B.2.11.

B.2.1 Object



Prompt:
You are a ResponseEditorGPT who is given an instruction and a response generated by a vision-language model. The instruction asks about the objects, or actions. Your task is as follows:

1. Modify the "objects" or "action" to make the "Original Response" about "objects" or "actions" incorrect.

2. "New Response" must be linguistically very similar to "Original Response" and must be incorrect.

3. You must ensure changes must be realistic given world knowledge.

4. You can minimally change other spans of the sentence to grammatical correctness and fluency.

5. The output format should be "New Response:"

Instruction: What type of flooring is in the room?

Original Response: The room has hard wood floors.

Your Turn
New Response:

Chosen Response: The room has hard wood floors.

Hard-negative Rejected Response: The room has carpeted floors.

Figure 7: Example prompt and generated hard-negative for VAPR sample of object task category.

Instruction: What type of airplane is displayed in the image?

Chosen Response: The image displays a World War era airplane with a propeller hanging in a museum.

Rejected Response: The image displays a World War era airplane with jet engines hanging in a museum.

(a) Task requiring perception of an object's components

Instruction: What is the woman doing in the image?

Chosen Response: The woman is sitting at a desk or table, working on a laptop computer.

Rejected Response: The woman is standing next to a desk or table, working on a laptop computer.

(b) Task requiring perception of an object's action

Instruction: What type of cake is on the plate?

Chosen Response: There is a slice of chocolate cake on the plate.

Rejected Response: There is a slice of strawberry cake on the plate.

(c) Task requiring perception of an object's type

Instruction: What is the cat doing in the image?

Chosen Response: The cat is sitting on the edge of a bathroom sink.

Rejected Response: The cat is sleeping on the edge of a bathroom sink.

(d) Task requiring perception of an object's action

Figure 8: Qualitative examples of the object task category, like object perception, its type, components, and associated actions.

B.2.2 Color

Prompt:
You are a ResponseEditorGPT who is given an instruction and a response generated by a vision-language model. The instruction asks about the colors of objects, environments, or themes. Your task is as follows:

1. Modify the "colors" of all objects, environments, or themes in the response to make "Original Response" about "colors" incorrect.

2. You must only change the "colors", so that "New Response" is linguistically very similar to "Original Response" and is incorrect.

3. The "New colors" you use to replace original colors must be unique and not be too descriptive.

4. The "New colors" must be realistically possible, considering the object they describe.

5. You cannot use colors in the penalty list.

6. You can minimally change other spans of the sentence to grammatical correctness and fluency.

7. List the "New colors" you replace within the response.

Penalty list: [white, blue]

Instruction: What color is the kitchen counter where the vegetables are placed?

Original Response: The kitchen counter where the vegetables are placed is green.

Your Turn
New Response:
New Colors:

Chosen Response: The kitchen counter where the vegetables are placed is green.

Hard-negative Rejected Response: The kitchen counter where the vegetables are placed is yellow.

New Colors: [yellow]

Figure 9: Example prompt and generated hard-negative for VAPR sample of color task category. Here, the penalty list is simulated for illustration. New colors represent additional information generated alongside the hard-negative rejected response to maintain the penalty list. We maintain a list of most perturbed color values (size 2), updated after every 10 hard-negative rejected response generations. Note, if more than one color is perturbed, "New Colors" will have more elements in the list.

Instruction: What color are the couches in the living room?

Chosen Response: The couches in the living room are black.

Rejected Response: The couches in the living room are orange.

(a) Task requiring color perception of an object.

Instruction: What are the colors of the flowers in the vase?

Chosen Response: The colors of the flowers in the vase are red, green, and purple.

Rejected Response: The colors of the flowers in the vase are yellow, blue, and orange.

(b) Task requiring color perception of mulitple objects.

Instruction: What is the color of the surfboard the dog is sitting on?

Chosen Response: The color of the surfboard the dog is sitting on is blue.

Rejected Response: The color of the surfboard the dog is sitting on is orange.

(c) Task requiring color perception of an object.

Instruction: What is the overall color theme of the living room?

Chosen Response: The overall color theme of the living room is predominantly white, with white furniture and white elements on the brick wall.

Rejected Response: The overall color theme of the living room is predominantly turquoise, with turquoise furniture and turquoise elements on the brick wall.

(d) Task requiring color perception of the theme of a background setting.

Figure 10: Qualitative examples of the color task category, like perception of colors of a single object, multiple objects and background theme.

B.2.3 Size

Prompt:
You are a ResponseEditorGPT who is given an instruction and a response generated by a vision-language model. The instruction asks about the "size" of objects or themes. Your task is as follows:

1. Modify the "size" of the objects or themes to make the "Original Response" about the "size" incorrect.

2. "New Response" must be linguistically very similar to "Original Response" and must be incorrect.

3. You must ensure changes must be realistic given world knowledge.

4. You can minimally change other spans of the sentence to grammatical correctness and fluency.

5. The output format should be "New Response:"

Instruction: How is the cheeseburger described in terms of size and ingredients?

Original Response: The cheeseburger is described as massive and containing double cheese layers.

Your Turn
New Response:

Chosen Response: The cheeseburger is described as massive and containing double cheese layers.

Hard-negative Rejected Response: The cheeseburger is described as small and containing a single cheese layer.

Figure 11: Example prompt and generated hard-negative for VAPR sample of size task category.

Instruction: What is the general size of the room?

Chosen Response: The room is described as a small living area, which implies that it is not very large or spacious.

Rejected Response: The room is described as an expansive living area, which implies that it is quite large and spacious.

(a) Task requiring thematic size perception, here room.

Instruction: How would you describe the size of the group of people flying kites?

Chosen Response: The group of people flying kites is large, indicating that it is a popular and well-attended event or gathering.

Rejected Response: The group of people flying kites is small, indicating that it is a more intimate and possibly less publicized event or gathering.

(b) Task requiring thematic size perception, here gathering of people.

Instruction: What's the size of the refrigerator in the image?

Chosen Response: The refrigerator in the image is a small or mini-sized refrigerator.

Rejected Response: The refrigerator in the image is a large, full-sized refrigerator.

(c) Task requiring size perception of an object.

Instruction: What is the size of the cake in the image?

Chosen Response: The cake in the image is quite large and tall, comprising multiple layers.

Rejected Response: The cake in the image is small and flat, consisting of a single layer.

(d) Task requiring size perception of an object.

Figure 12: Qualitative examples of the size task category, like size perception of an object or broadly a theme (eg. room, gathering of people)

B.2.4 Background

Prompt:
You are a ResponseEditorGPT who is given an instruction and a response generated by a vision-language model. The instruction asks about the "time", "weather", or "environment" of events, surroundings, or themes. Your task is as follows:

1. Modify the "time", "weather", or "environment" of events, surroundings, or themes to make the "Original Response" about the "time", "weather", or "environment" incorrect.

2. "New Response" must be linguistically very similar to "Original Response" and must be incorrect.

3. You must ensure changes must be realistic given world knowledge.

4. You can minimally change other spans of the sentence to grammatical correctness and fluency.

5. The output format should be "New Response:"

Instruction: What time of day is it in the image?

Original Response: It is nighttime in the image, as evidenced by the dark sky background.

Your Turn
New Response:

Chosen Response: It is nighttime in the image, as evidenced by the dark sky background.

Hard-negative Rejected Response: It is twilight in the image, as evidenced by the darkening sky background.

Figure 13: Example prompt and generated hard-negative for VAPR sample of background task category. Notably, the slight linguistic change from "dark" to "darkening" has significant change in semantics.

Instruction: What's the weather like in the image?

Chosen Response: The weather in the image is rainy, creating a wet atmosphere on the street.

Rejected Response: The weather in the image is snowy, creating a cold atmosphere on the street.

(a) Task requiring perception of weather.

Instruction: What kind of environment is the giraffe in?

Chosen Response: The giraffe is in a dry, arid environment with brown vegetation, scrub bushes, and sand in the field. It appears to be a somewhat harsh, wild habitat that the giraffe is navigating.

Rejected Response: The giraffe is in a lush, green environment with dense vegetation, leafy bushes, and a grassy field. It appears to be a somewhat fertile, wild habitat that the giraffe is navigating.

(b) Task requiring perception of environment.

Instruction: What type of lighting is featured in the living room?

Chosen Response: The living room features spot lighting, which provides a focused illumination on specific areas or objects within the room.

Rejected Response: The living room features ambient lighting, which provides a general, diffused illumination throughout the room.

(c) Task requiring perception of background lighting.

Instruction: What are the weather conditions in the image?

Chosen Response: The weather conditions in the image are rainy, as evidenced by the group of people holding umbrellas to protect themselves from the pouring rain.

Rejected Response: The weather conditions in the image are rainy, as evidenced by the group of people holding umbrellas to protect themselves from the light drizzle.

(d) Task requiring perception of weather.

Figure 14: Qualitative examples of the background task category, like perception of weather, environment, lighting, and time of the day.

B.2.5 Counting

Prompt:
You are a ResponseEditorGPT who is given an instruction and a response generated by a vision-language model. The instruction asks about the counts of objects. Your task is as follows:

1. Modify the "counts" of all objects in the response to make "Original Response" about "counts" incorrect.

2. You must only change the "counts", so that "New Response" is linguistically very similar to "Original Response" and is incorrect.

3. The "New counts" you use to replace original colors must be unique and not be too descriptive.

4. The "New counts" must be realistically possible, considering the object they describe.

5. You cannot use counts in the penalty list, neither the word form in the penalty nor its numerical form.

6. You can minimally change other spans of the sentence to grammatical correctness and fluency.

7. List the "New counts" you replace within the response.

Penalty list: [two, seven]

Instruction: How many people are visible near the truck in the image?

Original Response: There are four people visible near the truck in the image. Three people are standing in front of the truck, while another person is in the background.

Your Turn
New Response:
New Colors:

Chosen Response: There are four people visible near the truck in the image. Three people are standing in front of the truck, while another person is in the background.

Hard-negative Rejected Response: There are six people visible near the truck in the image. Five people are standing in front of the truck, while another person is in the background.

New Counts: [six, five]

Figure 15: Example prompt and generated hard-negative for VAPR sample of counting task category. Here, the penalty list is simulated for illustration. New counts represent additional information generated alongside the hard-negative rejected response to maintain the penalty list. Similar to color task category, We maintain a list of most perturbed count values (size 2), updated after every 10 hard-negative rejected response generations. Note, if more than one count is perturbed, "New Counts" will have more elements in the list.

Instruction: How many motorcycles are in the image?

Chosen Response: There are three motorcycles in the image.

Rejected Response: There are six motorcycles in the image.

(a) Task requiring counting capability.

Instruction: How many bears are present in the image?

Chosen Response: There are three bears present in the image - an adult bear and two bear cubs.

Rejected Response: There are seven bears present in the image - an adult bear and six bear cubs.

(b) Task requiring counting capability.

Instruction: How many women are in the image holding teddy bears?

Chosen Response: There are three women in the image holding teddy bears.

Rejected Response: There are two women in the image holding teddy bears.

(c) Task requiring counting ability.

Instruction: How many road signs are there in the image?

Chosen Response: There are several road signs in the image, including two One Way signs, one of which is upside down.

Rejected Response: There are no road signs in the image, and the streets are empty of any directives.

(d) Task requiring counting capability.

Figure 16: Qualitative examples of the counting task category.

B.2.6 Spatial Reasoning

Prompt:
You are a ResponseEditorGPT who is given an instruction and a response generated by a vision-language model. The instruction asks about the "spatial relation", of objects. Your task is as follows:

1. Modify the "spatial relation" of objects to make the "Original Response" about the "spatial relation" incorrect.

2. "New Response" must be linguistically very similar to "Original Response" and must be incorrect.

3. You must ensure changes must be realistic given world knowledge.

4. You can minimally change other spans of the sentence to grammatical correctness and fluency.

5. The output format should be "New Response:"

Instruction: Where is the toilet located in the image?

Original Response: The toilet is located outdoors, surrounded by a field of grass and trees, in the middle of the woods.

Your Turn
New Response:

Chosen Response: The toilet is located outdoors, surrounded by a field of grass and trees, in the middle of the woods.

Hard-negative Rejected Response: The toilet is located indoors, surrounded by white-tiled walls and a sink, in the corner of a bathroom.

Figure 17: Example prompt and generated hard-negative for VAPR sample of spatial reasoning task category.

Instruction: What position is the cat in while laying on the rug?

Chosen Response: The cat is laying on its back on the rug.

Rejected Response: The cat is laying on its side on the rug.

(a) Task requiring instance location.

Instruction: Where is the person riding the bicycle in the image?

Chosen Response: The person is riding the bicycle on a city street, specifically in a bicycle lane near many street signs.

Rejected Response: The person is riding the bicycle on a dirt path in a park, far away from any street signs.

(b) Task requiring instance location.

Instruction: How are the motorcycles arranged in the image?

Chosen Response: The motorcycles are arranged in rows or parked together in a row, which creates an organized and neat appearance.

Rejected Response: The motorcycles are arranged in a circle, with each facing outwards, which creates a symmetrical and organized appearance.

(c) Task requiring relative spatial reasoning.

Instruction: What is the position of the skier's ski poles?

Chosen Response: The skier has tucked his ski poles under his arms while racing through the snow.

Rejected Response: The skier is holding his ski poles parallel on either side, with each pole pointing outward from his body as he navigates through the snow.

(d) Task requiring relative spatial reasoning.

Figure 18: Qualitative examples of the spatial reasoning task category, involve instance location or relative spatial reasoning.

B.2.7 Existence

Prompt:

You are a ResponseEditorGPT who is given an instruction and a response generated by a vision-language model. The instruction asks about an "existence" of an object, object attribute, object count, object spatial relation, object comparison, background or theme. Your task is as follows:

1. Modify the original response to change the polarity of the response, that is, make "Yes" a "No" and "No" a "Yes".

2. Paraphrase both the "Original Response" and the "New Response", such that it says, "Yes" or "No" followed by the ask in the question.

3. "New Response" must be linguistically very similar to "Original Response" and must be incorrect.

4. You must ensure changes must be realistic given world knowledge.

5. You can minimally change other spans of the sentence to grammatical correctness and fluency.

6. The output format should be "Original Response: " "New Response

Instruction: Would this man score a touchdown?

Original Response: No, the man would not score a touchdown.

Your Turn
New Response:

Chosen Response: No, the man would not score a touchdown..

Hard-negative Rejected Response: Yes, the man would score a touchdown.

Figure 19: Example prompt and generated hard-negative for VAPR sample of object task category.

Instruction: Are there any people in the picture?

Chosen Response: No, there are no people shown in the picture.

Rejected Response: Yes, there are people visible in the image.

(a) Task requiring object perception

Instruction: Are the batters ankles twisted?

Chosen Response: Yes, the batter's ankles are twisted.

Rejected Response: No, the batter's ankles are not twisted.

(b) Task requiring perception of an object's state and action

Instruction: Does the elbow pad to the right of the other elbow pad have black color?

Chosen Response: No, the elbow pad to the right of the other elbow pad does not have a black color.

Rejected Response: Yes, the elbow pad to the right of the other elbow pad has a black color.

(c) Task requiring color perception spatial reasoning

Instruction: Does the light say it is ok to walk?

Chosen Response: Yes, the light indicates it's safe to walk.

Rejected Response: No, the light says it's not safe to walk.

(d) Task requiring object perception and knowledge

Figure 20: Qualitative examples of the existence task category.

B.2.8 Referential VQA

Prompt:

You are a ResponseEditorGPT who is given an instruction and a response generated by a vision-language model. The instruction asks about counts, color, spatial location, comparison or existence of objects. More than one of the tasks can be asked in an instruction. Your task is as follows:

1. Identify the different tasks asked in the question. You do not have to output this, only understand the intent.

2. Modify the spans in the response which answer the different tasks in the instruction to make "Original Response" incorrect.

3. You must only change the "spans", so that "New Response" is linguistically very similar to "Original Response" and is incorrect, while maintaining rest of the response.

4. You can minimally change other spans of the sentence to semantic correctness, grammatical correctness and fluency.

5. If the task is about colors or counts, ensure that you change the span with wide range of colors and counts respectively.

6. The "New colors" or "New Counts" must be realistically possible, considering the object they describe.

7. If the response is one word or small phrase, paraphrase both the "Original Response" and the "New Response", such that it says "New Response" is incorrect with respect to the "Original Response" while being semantically sensible. Both "Original Response" and "New Response" must now be full sentences.

Instruction: What is the size difference between these two boats?

Original Response: There is a noticeable size difference between the two boats, with one being considerably larger than the other smaller boat.

Your Turn
New Response:

Chosen Response: There is a noticeable size difference between the two boats, with one being considerably larger than the other smaller boat.

Hard-negative Rejected Response: There is no noticeable size difference between the two boats, as they appear to be identical in size.

Figure 21: Example prompt and generated hard-negative for VAPR sample of referential VQA task category.

Instruction: What is the size of the dog compared to the child?

Chosen Response: The dog is described as large in comparison to the child. This implies that the dog may be of a bigger breed or perhaps a fully-grown adult dog.

Rejected Response: The dog is described as small in comparison to the child. This implies that the dog may be a smaller breed or perhaps still a puppy.

(a) Task requiring object perception

Instruction: How many people are in the image and where are they located?

Chosen Response: There are two people, a man and a woman, in the image, and they are located in a kitchen.

Rejected Response: There are five people, consisting of three men and two women, in the image, and they are located in a living room.

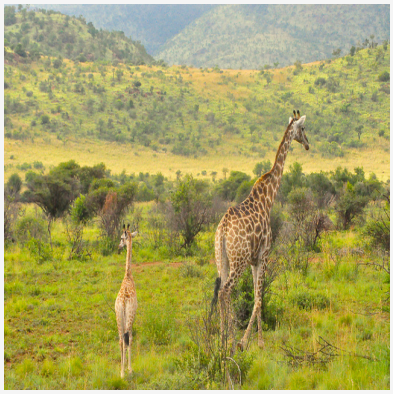
(b) Task requiring perception of an object's state and action

Instruction: How many colors are the cows in the image?

Chosen Response: There are three main colors of cows in the image: black, brown, and white.

Rejected Response: There are four main colors of cows in the image: black, gray, tan, and white.

(c) Task requiring color perception spatial reasoning



Instruction: How many giraffes are visible in the image, and what are their relative sizes?

Chosen Response: There are two giraffes visible in the image: a large adult giraffe and a smaller kid giraffe, likely its offspring.

Rejected Response: There are three giraffes visible in the image: a medium-sized adult giraffe along with two smaller giraffes, possibly its offspring.

(d) Task requiring object perception and knowledge

Figure 22: Qualitative examples of the existence task category.

B.2.9 Image Captioning

Prompt:

You are a ResponseAnalyzerGPT who is given an instruction and a response generated by a vision-language model. The response will consist of one or more visual elements - objects, object relationships, object attributes, environment information or actions. Each visual element is modified by one or more dimension, where a dimension belongs to the set ("Color", "Number", "Size", "Shape", "Other object physical attribute", "Weather Time", "Background", "Spatial relationship", "Comparative relationship", "Other object relationship")
Your task is to list (visual element, dimension, phrase) triplet, where visual element is an element in the response, dimension modifies the visual element and phrase is a span from the response that shows how dimension modified the visual element.

You must follow the guidelines given below:

1. Do not repeat the same triplet.

2. The dimension must always belong to the set ("Color", "Number", "Size", "Shape", "Other object physical attribute", "Weather Time", "Background", "Spatial relationship", "Comparative relationship", "Other object relationship")

3. Output format Must be "Triplet List : []" where "[]" is a list of triplets

Instruction: Write a detailed description of the given image.

Original Response: The image features a unique scene of a green jetfighter airplane on display in an open area of the city. The airplane has white and pink accents painted on its design, making it visually striking. It is situated in the middle of the road, with tall buildings surrounding the area.

There are several people admiring and standing next to the green plane. Some of them can be found on the right side of the image, while another person is seen closer to the plane on the left side. In addition to the airplane, there is a truck parked nearby on the left side of the road.

Your Turn
Triplet List: []

Triplet List: [("jetfighter airplane", "color", "green jetfighter airplane"), ("airplane", "color", " white and pink accent"), ("it", "spatial relationship", "in the middle of the road"), ("buildings", "size", "tall buildings"),("people", "counting", "several people"),("some", "spatial relationship", " right side of the image"),("people", "spatial relationship", "standing next to the green plan"), ("person", "spatial relationship", " closer to the plane on the left side"), ("truck", "spatial relationship", " truck parked nearby on the left side of the road")]

Figure 23: Example prompt for extracting the different possible dimensions of perturbing an image caption (perception & reasoning). Note, we simulate the dimension extraction in this example.

Prompt:
You are a ResponseEditorGPT who is given an instruction and a response generated by a vision-language model. The response will consist of one or more visual elements - objects, object relationships, object attributes, environment information or actions. Each visual element is modified by one or more dimension, where a dimension must belong to the set ("Color", "Number", "Size", "Shape", "Other object physical attribute", "Weather Time", "Background", "Spatial relationship", "Comparative relationship", "Other object relationship"). You will also be given a list of (visual element, dimension, phrase) triplets, where visual element is an element in the response, dimension modifies the visual element and phrase is a span from the response that shows how dimension modified the visual element.

Your task is as follows:

1. For each triplet, modify the phrase in the "Original response" corresponding to each triplet along the dimension mentioned in the triplet to make the "Original response" incorrect.

2. "New Response" must be linguistically very similar to "Original response" and must be incorrect.

3. You must ensure changes must be realistic given world knowledge.

4. You can minimally change other spans of the sentence to grammatical correctness and fluency.

5. The output format should be "Original Response: " "New Response

Instruction: Write a detailed description of the given image.

Original Response: The image features a unique scene of a green jetfighter airplane on display in an open area of the city. The airplane has white and pink accents painted on its design, making it visually striking. It is situated in the middle of the road, with tall buildings surrounding the area.

There are several people admiring and standing next to the green plane. Some of them can be found on the right side of the image, while another person is seen closer to the plane on the left side. In addition to the airplane, there is a truck parked nearby on the left side of the road.

Triplet List: [("jetfighter airplane", "color", "green jetfighter airplane"), ("airplane", "color", " white and pink accent"), ("it", "spatial relationship", "in the middle of the road"), ("buildings", "size", "tall buildings"),("people", "spatial relationship", "standing next to the green plan"), ("person", "spatial relationship", " closer to the plane on the left side"), ("truck", "spatial relationship", " truck parked nearby on the left side of the road")]
Your Turn
New Response:

Chosen Response: The image features a unique scene of a green jetfighter airplane on display in an open area of the city. The airplane has white and pink accents painted on its design, making it visually striking. It is situated in the middle of the road, with tall buildings surrounding the area.

There are several people admiring and standing next to the green plane. Some of them can be found on the right side of the image, while another person is seen closer to the plane on the left side. In addition to the airplane, there is a truck parked nearby on the left side of the road.

Hard-negative Rejected Response: The image features a unique scene of a yellow jetfighter airplane on display in an open area of the city. The airplane has purple and orange accents painted on its design, making it visually striking. It is situated on the side of the road, with short buildings surrounding the area.

There are several people admiring and standing in the distance from the yellow plane. Some of them can be found on the right side of the image, while another person is seen farther from the plane on the left side. In addition to the airplane, there is a truck parked far from the right side of the road.

Figure 24: Example prompt for generating hard-negative rejected response for image captioning category. We randomly select 50% to 75% triplets and feed them to the prompt as shown above. Similar to colors and counting task categories, we maintain a list of most perturbed dimensions (size 2), updated after every 10 hard-negative rejected response generations. If a dimension is used more than once, its frequency is proportionately increased. Note image is the same as the previous prompt (see Fig. 23).

Instruction: Analyze the image in a comprehensive and detailed manner.

Chosen Response: The image presents an orange and white motor scooter with two passengers riding down the middle of a wet street. The person in the front is driving the scooter, while the person sitting behind them is holding a purple umbrella over both of them, providing protection from rain.

There are potted plants on the sidewalk, adding some greenery to the scene. Cars can be seen parked or driving in the background, sharing the road with the scooter. The scene depicts a typical rainy day on an urban street with vehicles and pedestrians going about their daily routine.

Rejected Response:
The image presents a black and white motor scooter with two passengers riding down the middle of a dry street. The person in the front is driving the scooter, while the person sitting behind them is holding a red umbrella over both of them, providing protection from the sun.

There are potted plants on the sidewalk, adding some greenery to the scene. Cars can be seen parked or driving in the background, sharing the road with the scooter. The scene depicts a typical sunny day on a suburban street with vehicles and pedestrians going about their daily routine.

(a) Task requiring color, object, and background perception.

Figure 25: Qualitative example of the image captioning task category.

B.2.10 General Reasoning

Prompt:

You are a ResponseEditorGPT who is given an instruction and a response generated by a vision-language model. The instruction asks about the "reasoning" of objects, events, environments, or themes. Your task is as follows:

1. Make the reasoning in the original response incorrect.

2. You can modify the objects, their attributes, related objected, or action and make the "original response" about "reasoning" is incorrect.

3. "New Response" must be linguistically very similar to "Original response" and must be incorrect.

4. You must ensure changes must be realistic given world knowledge.

5. You can minimally change other spans of the sentence to grammatical correctness and fluency.

6. The output format should be "Original Response: " "New Response

Instruction: What might be a possible reason for the man taking a picture of the dirt cake?

Original Response: A possible reason for the man taking a picture of the dirt cake could be that the cake is a unique and creative design, which features a construction scene on top. He might want to capture the design and decoration before it is served or share the photo with others to showcase the artistic and aesthetic aspects of the cake. Additionally, the image could be used as a memory of a special occasion or event for which the cake has been prepared.

Your Turn
New Response:

Chosen Response: A possible reason for the man taking a picture of the dirt cake could be that the cake is a unique and creative design, which features a construction scene on top. He might want to capture the design and decoration before it is served or share the photo with others to showcase the artistic and aesthetic aspects of the cake. Additionally, the image could be used as a memory of a special occasion or event for which the cake has been prepared.

Hard-negative Rejected Response: A possible reason for the man taking a picture of the dirt cake could be that the cake is a rare and delicate design featuring an underwater scene on top. He might want to capture the design and the aquatic elements before it is served or share the photo with others to highlight the intricate and sea-themed aspects of the cake. Additionally, the image could be used as a memory of a beach-themed event or occasion for which the cake has been prepared.

Figure 26: Example prompt and generated hard-negative for VAPR sample of general reasoning task category.

B.2.11 Examples Showing Generation Steps for Task Categories with Penalty List

In this section, we use the color task as an example to illustrate generation steps involving the penalty list (counting task will have a similar way of prompting). We use a penalty list of size K=10, updated every 10 samples, with one retry generation attempt possible. Note, until we have atleast K (here =10) distinct colors, the penalty list can be < K.

Stochastic Generation: Since generation is stochastic, the exact penalty list used for a given sample may vary. Therefore, we simulate several cases:

- **Case-1: Empty list** When the penalty list is empty.
- **Case-2** When the penalty list is non-empty, with sub-cases:
  - **Case-2a: First Attempt** The generation succeeds on the first attempt without conflicting with the penalty list.
  - **Case-2b: Second Attempt** The generation initially conflicts but succeeds on a retry attempt.
  - **Case-2c: Rerun Script** If both attempts fail due to conflicts, the sample is put back into the pool. This situation happens neglibible number of times and the solution is to re-run the script. When re-running, it typically encounters a new penalty list because:
    - Some samples may have already been processed and help seed the penalty list.
    - The script randomizes sample order, ensuring that failed samples do not repeatedly see the same penalty list.

For the purpose of illustration, we assume no generation failures by the LLM and K=10, when penalty list is not empty. The ✓rejected response (new response) is actually a sample from the dataset, and hence is kept as the correct response across scenarios. Now, we provide two examples showcasing different cases:

Prompt:
You are a ResponseEditorGPT who is given an instruction and a response generated by a vision-language model. The instruction asks about the colors of objects, environments, or themes. Your task is as follows:

1. Modify the "colors" of all objects, environments, or themes in the response to make "Original Response" about "colors" incorrect.

2. You must only change the "colors", so that "New Response" is linguistically very similar to "Original Response" and is incorrect.

3. The "New colors" you use to replace original colors must be unique and not be too descriptive.

4. The "New colors" must be realistically possible, considering the object they describe.

5. You cannot use colors in the penalty list.

6. You can minimally change other spans of the sentence to grammatical correctness and fluency.

7. List the "New colors" you replace within the response.

Penalty list:

Instruction: What colors are present on the subway train in the image?

Original Response: The subway train in the image is orange, blue, and silver.

Your Turn
New Response:
New Colors:

Original Response: The subway train in the image is orange, blue, and silver.

Case-1: Empty list

Penalty List: []

New Response: The subway train in the image is pink, turquoise, and white. ✓
New Colors: ['pink', 'turquoise', 'white'] ✓

Case-2a: First Attempt

Penalty List: ['yellow', 'black', 'beige', 'teal', 'green', 'burgundy', 'sepia', 'lavender', 'purple', 'orange']

New Response: The subway train in the image is pink, turquoise, and white. ✓
New Colors: ['pink', 'turquoise', 'white'] ✓

New Response: The subway train in the image is pink, black, white. ✗
New Colors: ['pink', 'black', 'white'] ✗
Reason: Color *black* conflicts with penalty list, retry.

Case-2b: Second Attempt

Penalty List: ['yellow', 'black', 'beige', 'teal', 'green', 'burgundy', 'sepia', 'lavender', 'purple', 'orange']

New Response: The subway train in the image is pink, turquoise, and white. ✓
New Colors: ['pink', 'turquoise', 'white'] ✓

New Response: The subway train in the image is yellow, green, white. ✗
New Colors: ['yellow', 'green', 'white'] ✗
Reason: Colors *yellow & green* conflict with penalty list, put sample back to un-annotated pool and rerun the script with un-annotated samples.

Case-2c: Rerun-script (penalty list updates

Penalty List: ['yellow', 'black', 'beige', 'blue', 'green', 'red', 'silver', 'lavender', 'purple', 'orange']

New Response: The subway train in the image is pink, turquoise, and white. ✓
New Colors: ['pink', 'turquoise', 'white'] ✓

New Response: The subway train in the image is red, turquoise, white. ✗
New Colors: ['red', 'turquoise', 'white'] ✗
Reason: Color *red* conflicts with penalty list, go to Case-2b.

Prompt:
You are a ResponseEditorGPT who is given an instruction and a response generated by a vision-language model. The instruction asks about the colors of objects, environments, or themes. Your task is as follows:

1. Modify the "colors" of all objects, environments, or themes in the response to make "Original Response" about "colors" incorrect.

2. You must only change the "colors", so that "New Response" is linguistically very similar to "Original Response" and is incorrect.

3. The "New colors" you use to replace original colors must be unique and not be too descriptive.

4. The "New colors" must be realistically possible, considering the object they describe.

5. You cannot use colors in the penalty list.

6. You can minimally change other spans of the sentence to grammatical correctness and fluency.

7. List the "New colors" you replace within the response.

Penalty list:

Instruction: What color are the bags or packages containing the office supplies?

Original Response: The office supplies are contained in small pink packages or a pink envelope.

Your Turn
New Response:
New Colors:

Original Response: The office supplies are contained in small pink packages or a pink envelope.

Case-1: Empty list

Penalty List: []

New Response: The office supplies are contained in small teal packages or a teal envelope. ✓
New Colors: ['teal'] ✓

Case-2a: First Attempt

Penalty List: ['purple', 'yellow', 'white', 'orange', 'green', 'brown', 'silver', 'red', 'pink', 'maroon']

New Response: The office supplies are contained in small teal packages or a teal envelope. ✓
New Colors: ['teal'] ✓

New Response: The office supplies are contained in small white packages or a white envelope. ✗
New Colors: ['white'] ✗
Reason: Color *white* conflicts with penalty list, retry.

Case-2b: Second Attempt

Penalty List: ['purple', 'yellow', 'white', 'orange', 'green', 'brown', 'silver', 'red', 'pink', 'maroon']

New Response: The office supplies are contained in small teal packages or a teal envelope. ✓
New Colors: ['teal'] ✓

New Response: The office supplies are contained in small orange packages or a transparent envelope. ✗
New Colors: ['orange', 'transparent'] ✗
Reason: Color *orange* conflicts with penalty list, even when transparent is reasonable and has no conflict. Put sample back to un-annotated pool and rerun the script with un-annotated samples.

Case-2c: Rerun-script (penalty list updates

Penalty List: ['yellow', 'gold', 'blue', 'black', 'brown', 'silver', 'red', 'pink', 'maroon']

New Response: The office supplies are contained in small teal packages or a teal envelope. ✓
New Colors: ['teal'] ✓

New Response: The office supplies are contained in small silver packages or a silver envelope. ✗
New Colors: ['silver'] ✗
Reason: Color *silver* conflicts with penalty list, go to Case-2b.

Table 4: Comparison of the VAPR Preference Dataset with Related Works: We compare VAPR to HA-DPO (Zhao et al., 2023), POVID (Zhou et al., 2024a), RLAIF-V (Yu et al., 2024b), and CSR (Zhou et al., 2024b) based on stylistic and length analysis, using linguistic similarity via word-level Levenshtein distance (LD) and token-level sequence length differences, respectively. We report the percentage of samples where chosen responses are longer $(chosen > rejected)$ or shorter $(rejected > chosen)$ for each overall, short, and long response category, with corresponding token differences following the same color notation. A lower LD stands for higher stylistic similarity, and lower token-level sequence length differences stand for higher length similarity. Note CSR only had long responses.

	Ours	HA-DPO	POVID	RLAIF-V	CSR
Overall Samples (%)	(21, 79)	(41, 59)	(24, 76)	(45, 55)	(38, 62)
- Linguistic Similarity	6	49	30	62	97
- Avg. Token Length Difference	3	18	16	15	27
- Token Length Difference	(10, 1)	(15, 20)	(27, 13)	(17, 14)	(23, 29)
Short Response Samples (%)	(19, 81)	(49, 51)	(10, 90)	(43, 57)	-
- Linguistic Similarity	3	24	16	17	-
- Avg. Token Length Difference	4	16	10	10	-
- Token Length Difference	(7, 1)	(14, 18)	(11, 10)	(14, 7)	-
Long Response Samples (%)	(33, 67)	(35, 65)	(46, 54)	(45, 55)	(38, 62)
- Linguistic Similarity	19	65	53	76	97
- Avg. Token Length Difference	8	20	27	18	27
- Token Length Difference	(17, 4)	(19, 21)	(32, 23)	(19, 17)	(23, 29)

B.3 Fine-grained length & Stylistic analysis

We analyze stylistic and length similarity between chosen and rejected responses, both overall and split by response length (see Table 4). We define responses with <=100 tokens as short and >100 tokens as long. Notably, even for long responses - where other methods exhibit higher dissimilarity and greater length differences - VAPR maintains higher linguistic similarity and lower token-level differences through its targeted response editing approach. This highlights the hard-negative nature of VAPR rejections and the nuanced distinctions its models are trained to optimize for.

B.4 Human Study

To evaluate the dataset, we conducted a human annotation study with three annotators per sample, including the authors of this paper. The setup and process are detailed as follows:

- **Sample Selection:**
  - A total of 500 samples were randomly stratified across ten task categories.
  - To prevent bias from task repetition, consecutive samples were ensured to come from different task categories.
- **Annotation Procedure:**
  - The study spanned two days, with 250 annotations collected per day.
  - Each day consisted of four hours of annotation, divided into one-hour sessions, with a 30-minute break after each session to mitigate annotator fatigue.
  - Annotators were presented with the following elements for each sample:
    - An image.
    - An instruction.
    - A chosen response.
    - A rejected response.
  - The task required a binary annotation to determine whether the rejected response qualified as a *hard-negative* or was similar to the chosen response.
- **Annotation Criteria:**
  - Annotators were instructed to evaluate the content of the responses exclusively.
  - Factors such as response length or linguistic similarity were explicitly excluded, as these were analyzed in prior studies (see §3.4) to avoid the confounding factors of human subjectivity in length and linguistic preference.
- **Quality Assurance:**
  - Results indicated that 97% of the samples aligned with the hard-negative response criteria.
  - Inter-annotator agreement (IAA), calculated using Fleiss' kappa, was 86%, signifying a high level of consistency among annotators.

These findings demonstrate that the dataset achieves high quality and reliability, despite being synthetically generated.

C Experimental Setup

C.1 Training Setup

This section outlines the training setup and hyperparameters, summarized in Table 5. Consistent hyperparameters were applied across all VAPR models, including LLaVA-VAPR (7B and 13B), Qwen2VL-VAPR (2B and 7B) and Qwen2.5VL-VAPR (3B and 7B). A shallow hyperparameter search was conducted for the learning rate, with 1e-6 yielding optimal results; 1e-5 caused model forgetting, while 1e-7 was insufficient for effective learning. An effective batch size of 32 was selected for training efficiency. The models were trained for 5 epochs using two A100 GPUs.

Hyperparameter	**Value**
Learning rate	1e-6
Learning rate Scheduler	Cosine
Warmup Ratio	0.03 (LLaVA) & 0.1 (Qwen2VL & 2.5VL)
Batch size	32
Lora r	128
Lora alpha	256
DPO Loss	sigmoid
DPO β	0.1
Max Sequence Length	2048

Table 5: Overview of training hyperparameters

C.2 Evaluation Benchmarks & Setup

C.2.1 Benchmarks

Open-ended & Descriptive Benchmarks : This includes LLaVA^{W} (LLaVA-in-the-wild) (Liu et al., 2023c), ConTextual (ConT) (Wadhawan et al., 2024), & MM-VET (MMV) (Yu et al., 2023). LLaVA^{W} assesses open-world visual reasoning and description. ConTextual tests joint reasoning over embedded text and visual elements across diverse text-rich image scenarios. MM-VET evaluates how well LVLMs perform on tasks integrating core-VL capabilities like OCR, spatial reasoning, and math. We report the GPT-4 scores obtained using the respective LLM-as-a-judge prompts of LLaVA^{W}, ConTextual, and MM-VET.

Vision-Centric Benchmarks : This includes SEED^{I} (SEED Bench image split) (Li et al., 2023a), CV (CV Bench) (Tong et al., 2024) and MMStar (MMS) (Chen et al., 2024a). SEED & MMS comprehensively evaluate perception and reasoning. CV assesses counting, spatial reasoning, and comparative reasoning (depth & distance). We report the overall accuracy for each benchmark.

Academic & Math Reasoning: This category includes MathVista (MV) (Lu et al., 2023), testmini subset, which evaluates mathematical reasoning across diverse problem types, and MMMU (Yue et al., 2024), which tests college-level academic reasoning across domains such as physical sciences, social sciences, finance, etc. For MMMU and MathVista we report the overall accuracy.

Hallucination & Adversarial : This category includes POPE (Li et al., 2023d) and NaturalBench (Li et al., 2024a). POPE assesses object hallucination by testing scenarios of object presence and absence. NaturalBench measures visio-linguistic compositionality and vision-blind behavior (bias to provide identical answers regardless of the image) of LVLMs by pairing two questions with two similar images yielding different answers. For POPE, we report the overall F1 score across three categories. For NaturalBench, we provide overall accuracy, per-image accuracy across two questions, per-question accuracy across paired images, and group accuracy, reflecting the model's ability to answer all four image-question combinations correctly.

C.2.2 Evaluation Setup

Model Generation Settings : To ensure consistency and equitable comparison with prior works (Liu et al., 2023b; Zhou et al., 2024a;b; Zhao et al., 2023; Wang et al., 2024b; Sun et al., 2023; Yu et al., 2024b), we set the generation temperature to 0 and the number of beams to 1 for all base, prior work and VAPR models.

LLM-as-a-Judge : For benchmarks such as LLaVA-bench, ConTextual, and MM-Vet, which require OpenAI APIs for evaluation, we utilize GPT-4 to align with prior methodologies. It is important to note that GPT-4 versions evolve due to periodic updates. For instance, prior works may have used GPT-4-0314, which was deprecated in June 2024[1]. To ensure consistency, we fix the version to GPT-4-0613, the current stable release, and calculate scores for these benchmarks across all baseline and VAPR models. Further, MathVista utilizes GPT-4-Turbo calls for answer extraction. To minimize our costs, we instead used GPT-4o, which is a more performant OpenAI model that is significantly cheaper. This evaluation incurred a total cost of approximately $800 (in addition to $300 for data generation using GPT-4o).

Significance analysis For statistical significance analysis, we employ bootstrap resampling (Koehn, 2004), which involves randomly sampling 50% of each benchmark's data and evaluating model performance over 1,000 iterations for comparing two models. We assess each preference-tuned baseline and VAPR model against the base SFT model, reporting statistical significance when the win rate is $\geq$ 95% (p = 0.05). Notably, VAPR models demonstrate improvements over base SFT models, with most results achieving statistical significance. In contrast, while showing improvements, prior works fail to meet the 95% confidence threshold under the bootstrap test (except CSR-13B on the Pope dataset), as shown in Table 2.

[1] GPT4 Depracation History

Table 6: Performance comparison of LLaVA-v1.5-Instruct, Qwen2VL-Instruct, Qwen2.5VL-Instruct, and DPO models finetuned on VAPR at three data scales 3K, 10K 30K, on ten benchmarks. Higher scores indicate better performance across all benchmarks, with the highest score for each benchmark highlighted in **bold**. PFT size refers to the preference for fine-tuning dataset size, where "-" represents no additional dataset. Each model is trained under identical hyperparameter settings.

Row	Method	PFT	LLaVA^{W}	ConT	MMV	SEED^{I}	CV	MV	MMMU	MMS	POPE	NB
1	LLaVA-1.5-7B	-	64.8	16.8	30.9	66.2	62.1	30.1	35.4	32.6	85.9	12.7
2	+ VAPR DPO	3K	69.9	19.2	31.4	66.1	61.9	30.1	35.4	32.9	85.1	13.6
3	+ VAPR DPO	10K	74.4	20.2	32.3	66.4	62.3	30.4	35.6	34.0	85.2	14.0
4	+ VAPR DPO	30K	**76.2**	**20.6**	**32.9**	**66.7**	**62.9**	**30.8**	**35.7**	**34.7**	85.4	**14.5**
5	LLaVA-1.5-13B	30.7	72.3	18.6	36.7	68.2	62.5	30.7	**36.1**	33.8	86.0	14.9
6	+ VAPR DPO	3K	75.6	19.4	36.3	68.3	63.5	31.1	35.2	34.9	86.0	15.8
7	+ VAPR DPO	10K	78.9	20.3	37.0	68.4	64.2	31.8	35.6	35.3	86.2	17.4
8	+ VAPR DPO	30K	**80.5**	**21.2**	**37.3**	**68.7**	**64.6**	**32.3**	35.8	**35.6**	86.3	**18.2**
9	Qwen2VL-2B	-	83.2	27.7	53.3	73.6	66.5	**51.0**	38.7	43.4	86.5	24.3
10	+ VAPR DPO	3K	83.8	31.6	52.6	73.6	67.5	50.2	38.7	43.4	87.6	24.9
11	+ VAPR DPO	10K	84.3	33.2	53.4	73.8	68.3	50.5	39.0	43.5	88.2	25.2
12	+ VAPR DPO	30K	**88.1**	**34.8**	**54.1**	**74.0**	**69.0**	50.9	**39.2**	**43.7**	**88.3**	**25.7**
13	Qwen2VL-7B	-	92.5	39.7	62.1	76.4	75.7	57.5	**50.7**	56.7	**87.3**	30.8
14	+ VAPR DPO	3K	92.8	41.9	63.0	76.4	75.8	57.2	50.6	57.1	87.0	31.7
15	+ VAPR DPO	10K	93.6	42.3	64.4	76.5	76.0	57.4	50.5	57.5	87.2	32.0
16	+ VAPR DPO	30K	**96.2**	**43.9**	**65.4**	**76.8**	**76.3**	**58.2**	50.0	**57.8**	**87.3**	**32.5**
17	Qwen2.5VL-3B	-	98.1	37.2	67.3	75.0	71.5	52.5	**45.7**	54.7	86.3	25.4
18	+ VAPR DPO	3K	95.1	38.0	66.8	75.0	71.6	52.5	45.3	55.0	86.0	25.5
19	+ VAPR DPO	10K	96.5	39.3	66.9	75.3	72.0	52.7	45.1	55.6	86.1	25.7
20	+ VAPR DPO	30K	97.1	**40.3**	**67.4**	**75.5**	**72.7**	**53.4**	44.9	**56.1**	**86.4**	**26.3**
21	Qwen2.5VL-7B	-	101.4	53.3	71.0	77.7	80.1	58.6	**50.9**	61.9	86.3	32.0
22	+ VAPR DPO	3K	100.6	53.1	71.2	77.6	80.5	58.6	**50.9**	62.0	86.4	32.0
23	+ VAPR DPO	10K	100.8	53.2	71.8	77.7	80.7	58.8	50.8	62.2	86.7	32.2
24	+ VAPR DPO	30K	**101.5**	**53.4**	**72.4**	**77.8**	**81.1**	**59.8**	50.6	**62.5**	**86.9**	**32.8**

D Extended Results

D.1 Data Scaling

D.2 Benchmark detailed results

We provide the breakdown of model performance across different task categories for two comprehensive dataset MMStar and CV Bench.

Table 7: **MMStar**: Performance is compared across LLaVA-v1.5-Instruct, Qwen2VL-Instruct, Qwen2.5VL-Instruct, and DPO models - preference finetuned on VaPR (30K subset created with GPT-4o). Higher scores indicate better performance, with the top result for each benchmark shown in bold. All models share the same hyperparameters. Abbreviations: CP = Coarse Perception, FGP = Fine-grained Perception, IR = Instance Reasoning, LR = Logical Reasoning, S&T = Science & Technology.

Model	Final Score	CP	FGP	IR	LR	S&T	Math
LLaVA-v1.5-Instruct-7B	32.6	58.8	26.8	40.0	26.0	17.2	26.8
VaPR-LLaVA-7B	**34.7**	**62.0**	**27.2**	**44.0**	**27.2**	**18.0**	**30.0**
LLaVA-v1.5-Instruct-13B	33.8	58.0	27.2	42.4	26.4	21.2	**27.6**
VaPR-LLaVA-13B	**35.6**	**60.8**	**28.4**	**47.6**	**28.0**	**23.6**	25.2
Qwen2VL-Instruct-2B	43.4	52.4	41.6	**51.6**	**43.2**	31.2	**40.4**
VaPR-Qwen2VL-2B	**43.7**	**53.6**	**45.6**	50.0	42.0	**31.6**	39.6
Qwen2VL-Instruct-7B	56.7	67.2	50.4	62.8	56.4	46.4	57.2
VaPR-Qwen2VL-7B	**57.8**	**67.6**	**51.2**	**63.2**	**57.6**	**49.2**	**58.0**
Qwen2.5VL-Instruct-3B	54.7	66.4	46.4	60.8	54.8	**39.6**	60.4
VaPR-Qwen2.5VL-3B	**56.1**	**68.4**	**47.2**	**63.6**	**55.2**	**39.6**	**62.4**
Qwen2.5VL-Instruct-7B	61.9	**72.0**	54.0	**70.8**	63.2	44.8	66.4
VaPR-Qwen2.5VL-7B	**62.5**	71.6	**55.2**	70.0	**64.4**	**45.6**	**68.4**

Table 8: **CV Bench**: Performance is compared across LLaVA-v1.5-Instruct, Qwen2VL-Instruct, Qwen2.5VL-Instruct, and DPO models - preference finetuned on VaPR (30K subset created with GPT-4o). Higher scores indicate better performance, with the top result for each benchmark shown in bold. All models share the same hyperparameters. Abbreviations: Overall = Overall Accuracy, Count = Count Accuracy, Spatial = Spatial Relation Accuracy, Depth = Depth (Order) Accuracy, Distance = Relative Distance Accuracy. Depth: Determine which of the two distinct objects is closer to the camera, and Relative Distance: Determine which of the two distinct objects is closer to the anchor object.

Model	Overall	Count	Spatial	Depth	Distance
LLaVA-v1.5-Instruct-7B	62.2	54.3	**71.2**	70.0	**56.3**
VaPR-LLaVA-7B	**62.9**	**57.2**	70.8	**71.7**	54.3
LLaVA-v1.5-Instruct-13B	62.5	58.8	68.2	69.7	55.8
VaPR-LLaVA-13B	**64.6**	**58.9**	**69.9**	**72.7**	**59.7**
Qwen2VL-Instruct-2B	66.5	66.6	67.5	65.8	67.5
VaPR-Qwen2VL-2B	**69.0**	**68.2**	**68.3**	**68.5**	**72.2**
Qwen2VL-Instruct-7B	75.7	**67.0**	79.5	**85.5**	72.8
VaPR-Qwen2VL-7B	**76.3**	66.6	**81.9**	82.7	**76.3**
Qwen2.5VL-Instruct-3B	71.5	68.5	74.8	78.2	66.3
VaPR-Qwen2.5VL-3B	**72.7**	**68.9**	**75.1**	**79.3**	**69.2**
Qwen2.5VL-Instruct-7B	80.1	68.5	90.0	86.7	78.5
VaPR-Qwen2.5VL-7B	**81.1**	**69.2**	**90.6**	**86.8**	**81.2**

From Tables 2,7 & 8, we observe that VaPR models consistently improve on perception tasks (particularly fine-grained perception), and reasoning tasks, such as spatial relationships (even complex ones like distance and depth) and counting. These gains align with their improved

performance on SeedBench and NaturalBench, both of which emphasize visio-linguistic compositionality, perception, and reasoning.

Notably, despite not being explicitly trained on OCR, textual reasoning, or math tasks, VaPR models achieve strong performance in these areas. We attribute this to their enhanced fine-grained perception, spatial reasoning, and counting capabilities, which demonstrates that improvements in these areas support interpretation of embedded text (consistent with prior work (Fu et al., 2024)) and geometric figures. This trend is further corroborated by gains on ConTextual and MathVista benchmarks. Interestingly, VaPR-Qwen2VL-2B achieves the largest gains on Pope, which can be explained by pronounced improvements in fine-grained perception (as evident in MMStar). On the other hand, it shows slight degradation in math, logical reasoning tasks, which can explain why it does not improve on MathVista, where the other models do (see Table 2).

Lastly, prior work (Ivison et al., 2024a) indicates that preference optimization primarily enhances truthfulness and alignment, rather than factuality, which aligns with our observed improvements in perception and reasoning but not in purely knowledge-based tasks. We hypothesize that limited gains in MMMU (see Table 2) can be attributed to the alignment tax VaPR models possibly pay in knowledge based tasks.

D.3 Preference Dataset Comparison

D.3.1 Training Setup

We compare VAPR with two alternative preference data generation techniques: (1) **POVID**, which uses GPT-4V to generate rejected responses given the image, instruction, and ground truth response, and (2) **SIMA**, which follows a self-preference generation paradigm by synthesizing two responses - one via greedy decoding and another via sampling (temperature = 1) - and using the same VLM as a critic to select the preferred response. For SIMA, we apply the generation method to the 10K subset of VAPR dataset; for POVID, we use the publicly released data (17K). For VAPR we select the 10K subset for training to ensure a fair comparison. We preference finetune LLaVA-v1.5 and Qwen2VL models on the above datasets. To ensure consistency, all models are trained for 50K steps, under identical hyperparameter settings. This setup enables a balanced and computationally efficient comparison, allowing us to evaluate all methods under a fixed training budget without an excessive number of experiments.

Table 9: Performance comparison of LLaVA-v1.5, Qwen2VL-Instruct, DPO models finetuned on VAPR, SIMA & POVID across 2B, 7B, and 13B parameter sizes on ten benchmarks. Higher scores indicate better performance across all benchmarks, with the highest score for each benchmark highlighted in **bold**. PFT size refers to the preference for fine-tuning dataset size, where **"-"** represents no additional dataset.

Row	Method	PFT	LLaVA^W	ConT	MMV	SEED^I	CV	MV	MMMU	MMS	POPE	NB
1	LLaVA-1.5-7B	-	64.8	16.8	30.9	66.2	62.1	30.1	35.4	32.6	**85.9**	12.7
2	+ VAPR-SIMA + DPO	10K	68.5	17.2	31.6	66.0	60.9	29.4	35.2	32.7	85.2	12.9
3	+ Povid DPO	17K	67.2	18.0	30.9	66.1	61.6	30.1	35.4	33.3	**85.9**	13.2
4	+ VAPR DPO	10K	**74.4**	**20.2**	**32.3**	**66.4**	**62.3**	**30.4**	**35.6**	**34.0**	85.2	**14.0**
5	LLaVA-1.5-13B	-	72.3	18.6	36.7	68.2	62.5	30.7	**36.1**	33.8	86.0	14.9
6	+ VAPR-SIMA + DPO	10K	73.3	18.2	35.5	67.7	61.0	31.2	34.5	33.6	86.0	13.1
7	+ Povid DPO	17K	74.5	19.8	34.6	68.1	63.9	31.5	34.9	34.1	**86.2**	15.3
8	+ VAPR DPO	10K	**78.9**	**20.3**	**37.0**	**68.4**	**64.2**	**31.8**	35.6	**35.3**	**86.2**	**17.4**
9	Qwen2VL-2B	-	83.2	27.7	53.3	73.6	66.5	**51.0**	38.7	43.4	86.5	24.3
10	+ VAPR-SIMA + DPO	10K	81.6	29.2	49.1	73.4	66.7	50.0	38.8	43.1	86.8	23.4
11	+ Povid DPO	17K	82.7	30.7	49.3	73.7	67.1	50.2	38.9	43.2	87.1	23.6
12	+ VAPR DPO	10K	**84.3**	**33.2**	**53.4**	**73.8**	**68.3**	50.5	**39.0**	**43.5**	**88.2**	**25.2**
13	Qwen2VL-7B	-	92.5	39.7	62.1	76.4	75.7	**57.5**	**50.7**	56.7	87.3	30.8
14	+ VAPR-SIMA + DPO	10K	90.1	38.5	62.9	76.1	75.4	56.5	50.0	57.0	87.2	30.5
15	+ Povid DPO	17K	90.9	37.4	63.5	76.2	75.8	57.0	50.4	57.2	**87.6**	31.1
16	+ VAPR DPO	10K	**93.6**	**42.3**	**64.4**	**76.5**	**76.0**	57.4	50.5	**57.5**	87.2	**32.0**

D.3.2 Analysis

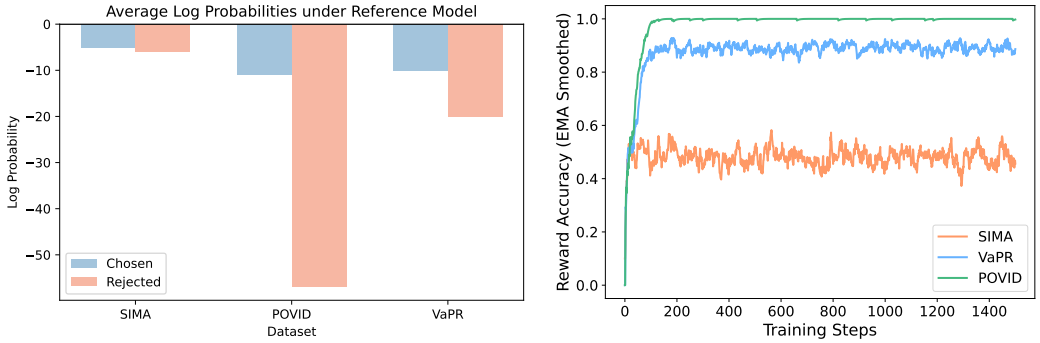
From Table 9, we observe that VAPR models consistently outperform both SIMA and POVID across model families. SIMA yields minimal to no improvements over base models and often degrades performance, especially for Qwen2VL. POVID achieves moderate gains for LLaVA but underperforms on Qwen2VL. On average, VAPR outperforms SIMA by 5-6% on LLaVA and 3-4% on Qwen2VL, and POVID by 4-5% and 2-3%, respectively. To understand these differences, we analyze DPO optimization using LLaVA-1.5-7B as a representative model.

The DPO loss can be re-written as:

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left(\alpha \left[\log \pi_\theta(y_w \mid x) - \log \pi_\theta(y_l \mid x) - \left(\log \pi_{\text{ref}}(y_w \mid x) - \log \pi_{\text{ref}}(y_l \mid x) \right) \right] \right), \tag{2}$$

where π_θ is the learned policy, π_{ref} is the fixed reference model, and α is the temperature scaling factor. Let:

$$\Delta_\theta = \log \pi_\theta(y_w \mid x) - \log \pi_\theta(y_l \mid x),$$
$$\Delta_{\text{ref}} = \log \pi_{\text{ref}}(y_w \mid x) - \log \pi_{\text{ref}}(y_l \mid x).$$



(a) Reference model log-probabilities

(b) Reward accuracy trends

Figure 27: Comparison of preference datasets. (a) Average reference model log-probabilities for chosen vs. rejected responses across VaPR, SIMA, and POVID - lower values indicate lower reference likelihood. (b) Reward accuracy trends over training steps show that SIMA improves gradually while POVID saturates quickly. Repeating the figure for ease of reference

the loss simplifies to: $\mathcal{L}_{\text{DPO}} = -\log \sigma\left(\alpha(\Delta_\theta - \Delta_{\text{ref}})\right)$.

For POVID, we observe that Δ_{ref} is already higher than VaPR while having a similar log probability of chosen response under the reference model (see Fig. 27a), indicating that POVID rejected responses are substantially less likely under the reference model as compared VaPR's. This can be attributed to the higher linguistic and stylistic differences in POVID preference pairs as compared to VaPR (see Table 1). From Fig. 27b, we observe that the POVID model rapidly attains a reward accuracy of 1, suggesting that the model may be overfitting based on preference signals derived from length and stylistic differences instead of content differences alone. In contrast, VaPR achieves its highest reward accuracy more gradually and does not converge to 1, indicating reduced overfitting due to exposure to more challenging preference pairs.

In contrast, for SIMA, we observe $\Delta_{\text{ref}} \approx 0$ on average (see Fig. 27a), indicating that chosen and rejected responses are often nearly identical. Manual inspection confirms this, with $\sim 20\%$ of pairs being exact duplicates and many others highly similar. When $\Delta_{\text{ref}} \approx 0$, the DPO loss is driven entirely by Δ_θ, removing the reference model's regularizing influence. This causes the optimizer to treat weak or noisy preference signals as informative, leading to overconfident updates based on superficial differences. Consequently, the model struggles to distinguish between responses - reflected in 50% reward accuracy (see Fig. 27b), potentially learning undesirable behaviors that harm downstream performance. Other self-preference methods, such as RLAIF-V and CSR, which employ multi-step generation and scoring procedures and generate preference pairs with greater stylistic and linguistic variation (see Table 1), can mitigate the above issue, but similar to POVID, they may also inadvertently exploit the stylistic and length biases.

On the other hand, VaPR explicitly controls for stylistic and length similarity when constructing preference pairs, ensuring that DPO learns from content-level differences. However, in low-resource settings where high-quality SFT data may be limited (as required by VaPR), our method can complement self-preference generation approaches. For instance, one could leverage high-confidence responses from methods like CSR or RLAIF-V as chosen responses, and apply the VaPR pipeline to generate targeted hard negative - creating high-quality preference pairs in an unsupervised manner. We view this as a promising direction for future work and invite the community to explore such hybrid strategies.

Table 10: Performance is compared across LLaVA-v1.5-Instruct, Qwen2VL-Instruct, Qwen2.5VL-Instruct, and DPO models, all preference finetuned on VaPR (10K subset created with GPT-4o) or VaPR-OS (8K samples generated by the Open-source model Qwen3-32b). Aside from the model used for generation, the framework and prompts are identical. All models share the same hyperparameters. Higher scores indicate better performance, with the top result for each benchmark shown in **bold**. In case two models get the top scores, both are **bolded**, otherwise top score is **bolded** and the second highest score(s) is underlined.

Row	Method	LLaVA^W	ConT	MMV	SEED^I	CV	MV	MMMU	MMS	POPE	NB
1	LLaVA-1.5-7B-Instruct	64.8	16.8	30.9	66.2	62.1	30.1	35.4	32.6	**85.9**	12.7
2	+ VaPR-OS DPO	73.3	18.7	32.1	66.3	**62.3**	30.2	**35.6**	33.7	83.6	13.9
3	+ VaPR DPO	**74.4**	**20.2**	**32.3**	**66.4**	**62.3**	**30.4**	**35.6**	**34.0**	85.2	**14.0**
4	Qwen2VL-2B-Instruct	83.2	27.7	53.3	73.6	66.5	**51.0**	38.7	43.4	86.5	24.3
5	+ VaPR-OS DPO	84.1	32.8	53.3	73.7	67.9	50.2	38.9	**43.5**	88.0	**25.2**
6	+ VaPR DPO	**84.3**	**33.2**	**53.4**	**73.8**	**68.3**	50.5	**39.0**	**43.5**	**88.2**	**25.2**
7	Qwen2.5VL-3B-Instruct	**98.1**	37.2	**67.3**	75.0	71.5	52.5	**45.7**	54.7	**86.3**	25.4
8	+ VaPR-OS DPO	95.7	39.0	65.4	**75.3**	**72.0**	52.4	45.5	55.4	86.2	**25.7**
9	+ VaPR DPO	96.5	**39.3**	66.9	**75.3**	**72.0**	**52.7**	45.1	**55.6**	86.1	**25.7**

D.4 Open-Source Editor data results

We preference finetuned LLaVA-v1.5-Instruct-7B, Qwen2VL-Instruct-2B and Qwen2.5-VL-Instruct-3B on VaPR-OS (VaPR Open source), where the difference between the models are two aspects: (a) is missing reasoning and captioning samples (1K each), reason shared below (b) uses an open-weight model for generating responses. Key findings include:

Dataset Quality: VaPR-OS rejected responses exhibit similar hard-negative properties to those generated by GPT-4o, with an average token length difference of 6 (compared to 3 in VaPR) and a Levenshtein distance of 10 (vs. 6 in VaPR).

Model Performance: From Table 10, we observe that models trained on the open-weight dataset achieve 99% of the performance of those generated with GPT-4o, with both consistently outperforming the baseline on most benchmarks. This is expected, as VaPR-OS samples overlap with VaPR and exhibit similar linguistic properties and average token length. These results highlight that the VaPR pipeline generalizes effectively and is not limited to closed-weight models.

Interestingly, we find that the contribution of captioning and abstract reasoning tasks to preference learning is limited, which is consistent with prior work (Lai et al., 2024) suggesting that complex tasks like reasoning may benefit stepwise decomposition of preference datasets. In our case, captioning and abstract reasoning tasks (e.g. Considering the presence of two clocks on the building, what purpose might this architectural design serve?), can be decomposed into simpler components like fine-grained perception (e.g., attribute recognition) and spatial reasoning (e.g., object location). Training models using these atomic tasks as step-wise preference samples may collectively support learning for more complex tasks, a direction we plan to investigate further.

Limitations and Future Directions for OS editors: We observed that data generated using Qwen3 sometimes fails to consistently perturb dependent spans (e.g., object perturbation: "... bathroom ... has a large bathtub ..." → "... kitchen ... has a large bathtub ...", whereas GPT-4o correctly changes it to "... kitchen ... has a large oven ..."), with this issue being prominent in captioning and abstract reasoning tasks. This could potentially add noise to the dataset and thus we omit these samples in our analysis. To mitigate this issue, we plan to experiment with more open-source models and generation of step-wise preference datasets.