# Supplementary Material for DEVA: Fine-tuning Multimodal Large Language Models for Visual Perception Tasks

Debasmit Das[1] Munawar Hayat[1] Fatih Porikli[1]
[1]Qualcomm AI Research
{debadas, hayat, fporikli}@qti.qualcomm.com

## 1. Use of Large Language Model

In this paper, we use GPT V only for refining and polishing the text in the paper.

## 2. Implementation Details

For datasets and benchmarking, we follow exact protocol and input prompts as introduced in [19] following guidelines [1]. A few-shot learning approach is considered for image classification and object detection task and for the rest we consider fine-tuning on a small-scale dataset. For the SFT-CoT dataset, we generate a CoT reasoning dataset using Qwen2.5-VL-32B-Instruct [1] with the input as the image and the prompt as the following:

```
Question: <QUERY>
Answer: <OUTPUT>
Generate reasoning: Explain step by step how to find the answer from the image.
```

Here, <QUERY> and <OUTPUT> is replaced by the corresponding query and output in the training dataset. The generated reasoning is the ground-truth answer that will be used to train the model using SFT.

For fine-grained image classification benchmark, we consider four datasets: Flower102 [21], Pets37 [22], FGVC-Aircraft [20] and Car196 [12]. For evaluation, we consider 1-shot, 2-shot, 4-shot, 8-shot and 16-shot protocol with the Qwen2-VL-2B model. For GRPO and their variants, we always use 8 generations. For all the datasets, we train for 8 epochs except for Pets37, where we train for 24 epochs. For training, we use a batch size of 8 distributed across 8 GPUs. We use bf16 datatype during the fine-tuning and gradient accumulation steps of 2. We follow the training setup in the lines of the description [2]. For different baselines used for comparison, we use their open-source implementation and report results for the best hyper-parameter configuration. For DEVA, we use the following hyper-parameters: $\gamma = 0.5$, the default weight on diversity $L_{\text{div}}$ and regularization loss $L_{\text{reg}}$ are both $1e-4$. The default values of $a$, $b$ and $c$ are 1.0, 0.0 and 2.0, respectively. For computing alignment features, we use BLIP-2 [14] as the feature extractor, where we use the output of the Q-Former as the alignment features. These are then used to compute the alignment volume.

We also apply a few-shot learning setup for the object detection task. Specifically, we selected 8 classes from the COCO dataset and vary the number of fine-tuning samples per class. This includes 1, 2, 4, 8, and 16 training samples per class. This is done to construct training sets with very limited data. For this setup, we finetune Qwen2-VL-2B, while we fine-tune the Qwen2-VL-7B for the 4-shot case. The mAP for the 8 classes is calculated and the average is reported. The eight classes taken from the COCO dataset includes: bus, train, fire hydrant, stop sign, cat, dog, bed, toilet. The hyper-parameters are the same as that of fine-grained classification task except that for the 7B model, we use 4 generations for computing GRPO instead of 8.

We also evaluate on the LISA grounding benchmark, where the task is to ground the relevant part of an image given a query and an image. For the LISA grounding dataset, we finetune both the Qwen2-VL-2B and the Qwen2-VL-7B on 239 training samples. After finetuning is done, the model is then evaluated on the test and validation split of the LISA grounding benchmark. The hyper-parameters are the same as fine-grained classification except that fine-tuning is done for 6 epochs and for the 7B model, we use 4 generations instead of 8 for computing GRPO.

[1]https://github.com/Liuziyu77/Visual-RFT/tree/main
[2]https://github.com/Liuziyu77/Visual-RFT/issues/97

For Table 1, we use the same hyper-parameters as fine-grained classification except that for evaluation on the COCO dataset, the model is fine-tuned for 2 epochs while for evaluation using the LVIS dataset, the model is fine-tuned for 4 epochs.

## 3. Additional Comparison Studies

Table 1. **Object Detection Results** First six columns show open vocabulary results on COCO dataset. We trained on 65 base categories and tested on 15 novel categories. Seventh and ninth column show few-shot results on LVIS [9] dataset of 6 rare categories. We conducted 10-shot experiments on 6 rare categories from the LVIS dataset. Eighth and tenth column shows open vocabulary object detection results on LVIS dataset. We trained on the 65 base categories of the COCO dataset and tested on the 13 rare categories of the LVIS dataset. The parenthesis in the last column of the first row are the results of GroudingDINO-B [17]. Best results are shown in bold and second best results are underlined.

| **Models** | $mAP_n$ | $mAP_b$ | $mAP_{all}$ | $mAP_n$ | $mAP_b$ | $mAP_{all}$ | $mAP$ | $mAP$ | $mAP$ | $mAP$ |
|---|---|---|---|---|---|---|---|---|---|---|
| Qwen2-VL-2B \| 7B \| \| 2B \| 7B | 9.8 | 6.0 | 6.7 | 26.3 | 17.5 | 19.2 | 4.0 | 2.7 | 15.4 | 15.7 (**23.9**) |
| + SFT | 13.6 | 7.8 | 8.9 | 25.7 | 17.5 | 19.0 | 10.0 | 7.6 | 27.6 | 24.0 |
| + SFT-CoT | 17.1 | 12.8 | 12.2 | 29.3 | 20.8 | 22.1 | 13.5 | 12.2 | 28.9 | 27.4 |
| + PPO [24] | 27.6 | 16.2 | 17.3 | 33.2 | 23.1 | 24.9 | 16.3 | 17.2 | 30.1 | 28.5 |
| + PAPO [27] | 32.2 | 21.3 | 24.6 | 37.0 | 27.9 | 28.3 | 22.4 | 22.1 | 35.2 | 32.3 |
| + DAPO [29] | 32.3 | 21.6 | 25.7 | 36.2 | 27.8 | 27.1 | 23.1 | 22.0 | 35.4 | 32.0 |
| + Dr GRPO [18] | 33.1 | 22.4 | 26.3 | 37.0 | 28.9 | 28.5 | 24.2 | 23.1 | 36.7 | 33.3 |
| + BNPO [28] | 32.0 | 21.9 | 25.8 | 37.2 | 27.5 | 28.1 | 24.6 | 24.0 | 37.5 | 34.1 |
| + GRPO-CARE [2] | 33.6 | 23.1 | 27.2 | 38.0 | 28.9 | 29.5 | 25.6 | 24.2 | 36.6 | 33.0 |
| + CPPO [15] | 33.2 | 24.2 | 27.1 | 39.1 | 29.5 | 30.3 | 24.0 | 24.3 | 37.2 | 34.2 |
| + GMPO [30] | 32.5 | 23.1 | 26.4 | 37.8 | 28.3 | 29.2 | 24.3 | 23.9 | 35.2 | 33.3 |
| + GSPO [31] | 34.6 | 25.2 | 28.2 | 39.6 | 30.1 | 31.2 | 25.3 | 25.2 | 37.6 | 35.5 |
| + ViRFT [19] | 31.3 | 20.6 | 22.6 | 35.8 | 25.4 | 27.4 | 19.4 | 20.7 | 33.8 | 30.4 |
| + **DEVA** (**Div.**) | 37.8 | 28.1 | 30.2 | 41.3 | 32.8 | 34.0 | 26.2 | 27.3 | 40.1 | 37.2 |
| + **DEVA** (**Div.** + **Explor.**) | 38.9 | 30.0 | 31.3 | 42.5 | 33.9 | 35.1 | 27.4 | 28.6 | 41.4 | 38.6 |
| + **DEVA** (**Div.** + **Explor.** + Align. **Vol.**) | <u>39.9</u> | <u>31.3</u> | <u>32.5</u> | <u>43.9</u> | <u>34.6</u> | <u>36.8</u> | <u>28.7</u> | <u>29.9</u> | <u>42.8</u> | <u>39.8</u> |
| + **DEVA** (**Div.** + **Explor.** + Align. **Vol.** + **Agg.**) | **41.9** | **32.3** | **33.3** | **45.0** | **35.7** | **38.1** | **30.1** | **32.0** | **43.9** | **41.2** |

In Table 1, we report additional results on the open-vocabulary setup. The goal of this setup is to understand whether reinforcement fine-tuning can aid in better generalization compared to supervised fine-tuning (SFT). Specifically, we finetune both Qwen2-VL-2B and Qwen2-VL-7B on 65 base categories and evaluate on 13 novel categories. We also evaluate on the base categories as well as combination of base and novel categories. As expected, we can see that when we apply DEVA on top of Visual-RFT, it produces an improvement of **+10-12 pts** improvement in mAP across novel categories, base categories and an aggregated set of categories. We can even outperform the highly competitive GSPO by **+ 5 pts** improvement in mAP.

We also evaluate the model trained on COCO on 13 rare categories of the LVIS dataset. This is shown in the eight and tenth column of the Table 1. DEVA essentially produces **+10 pts** improvement over Visual-RFT and **+5-6 pts** improvement over GSPO. Finally, we also evaluate 10-shot object detection performance within the LVIS dataset of 6 rare categories. We show similar improvement in performance compare to Visual-RFT and GSPO. To summarize, from Table 1, it is clear that DEVA is more effective for open vocabulary setup and can easily boost generalization capabilities.

## 4. Boosting Competitive Methods

In this section, we analyze whether DEVA can boost existing competitive methods. This is shown for fine-grained image classification dataset in Table 2 and for LISA reasoning grounding dataset in Table 3. In Table 2, we see that our proposed method can produce **+4-5 pts** improvement in accuracy when applied to existing RL algorithms. For reasoning grounding task in Table 2, we also observe similar trends, where our proposed framework can produce improvements upto **+6-7 pts** in IoU.

## 5. Effect of Low Rank Adaptation

For adapting the multi-modal model on small-scale fine-tuning data, our default setup is to finetune the whole model. In this section, we consider the situation where we finetune LoRA instead of fine-tuning the whole model. We consider different variations for LoRA. This includes changing ranks for LoRA and also the attachment points. The adaptors are attached on the $Q, K, V$ matrices in the transformer layers of the large language model (LLM) and/or vision encoder (VE). The results are shown in Table 4. From the results, we see that fine-tuning LoRA instead of full finetuning produces subpar performance, which is expected since LoRA modifies a very small subspace of the parameter space compared to full finetuning. As

Table 2. **Few-shot results on Fine-grained Classification dataset.** We evaluated four fine-grained image classification datasets when **DEVA** is added to existing RL algorithms.

| Model | 1-shot | 2-shot | 4-shot | 8-shot | 16-shot |
|---|---|---|---|---|---|
| Qwen2-VL-2B [26] | 56.0 | 56.0 | 56.0 | 56.0 | 56.0 |
| + PAPO [27] | 81.1 | 84.2 | 81.9 | 85.9 | 86.2 |
| + PAPO + **DEVA** | 86.4 | 89.4 | 87.3 | 91.4 | 91.6 |
| + DAPO [29] | 81.3 | 83.9 | 82.3 | 86.2 | 86.6 |
| + DAPO + **DEVA** | 86.5 | 89.1 | 87.4 | 91.5 | 91.8 |
| + GRPO-CARE [2] | 82.5 | 85.5 | 83.5 | 86.7 | 87.1 |
| + GRPO-CARE + **DEVA** | 87.4 | 90.4 | 88.5 | 92.2 | 92.6 |
| + CPPO [15] | 81.9 | 86.7 | 83.8 | 87.3 | 86.9 |
| + CPPO + **DEVA** | 87.1 | 91.5 | 89.0 | 92.7 | 92.4 |
| + GSPO [31] | 82.6 | 85.2 | 84.0 | 87.8 | 88.0 |
| + GSPO + **DEVA** | 87.8 | 90.6 | 89.3 | 93.2 | 93.4 |

Table 3. **Reasoning Grounding Results on LISA [13]**. We evaluated reasoning grounding results when **DEVA** is added to existing RL algorithms.

| Model | $\text{mIoU}_{\text{test}}$ | $\text{mIoU}_{\text{val}}$ | $\text{gIoU}_{\text{test}}$ | $\text{mIoU}_{\text{test}}$ | $\text{mIoU}_{\text{val}}$ | $\text{gIoU}_{\text{test}}$ |
|---|---|---|---|---|---|---|
| Qwen2-VL-2B \| 7B [26] | 26.9 | 30.1 | 25.3 | 40.4 | 45.2 | 38.0 |
| + PAPO [27] | 38.2 | 41.4 | 35.6 | 44.2 | 47.9 | 43.8 |
| + PAPO + **DEVA** | 44.6 | 47.7 | 42.0 | 50.3 | 54.4 | 50.1 |
| + DAPO [29] | 39.4 | 42.6 | 37.2 | 44.7 | 48.1 | 43.7 |
| + DAPO + **DEVA** | 45.8 | 48.9 | 43.3 | 51.2 | 54.8 | 49.9 |
| + GRPO-CARE [2] | 39.4 | 42.6 | 36.1 | 45.1 | 49.1 | 44.7 |
| + GRPO-CARE + **DEVA** | 45.6 | 48.7 | 42.3 | 51.3 | 55.2 | 50.2 |
| + CPPO [15] | 40.1 | 43.3 | 36.2 | 45.6 | 49.2 | 45.0 |
| + CPPO + **DEVA** | 46.3 | 49.4 | 42.8 | 51.9 | 55.6 | 50.6 |
| + GSPO [31] | 41.3 | 44.5 | 37.1 | 46.0 | 49.9 | 46.1 |
| + GSPO + **DEVA** | 47.2 | 50.3 | 43.4 | 52.3 | 56.2 | 51.4 |

expected, higher ranks for LoRA produces higher IoU since it closely approximates full fine-tuning. When LoRA is attached to both VE and LLM, visual perception capabilities for multimodal LLMs are enhanced better compared to attaching LoRA to either VE or LLM. Furthermore, results show that it is more effective to attach LoRA to LLM instead of VE. This might be because the LLMs are more responsible for multimodal reasoning tasks and need to be adapted to the specific visual grounding task. On the other hand, the VE is already capable in handling perception tasks. This empirical evidence has also been highlighted before in [6].

Table 4. **Reasoning Grounding Results on LISA [13]**. We evaluated reasoning rounding results when **DEVA** is added to existing RL algorithms and finetuned using LoRA [11].

| Model | $mIoU_{test}$ | $mIoU_{val}$ | $gIoU_{test}$ | $mIoU_{test}$ | $mIoU_{val}$ | $gIoU_{test}$ |
|---|---|---|---|---|---|---|
| Qwen2-VL-2B \| 7B [26] | 26.9 | 30.1 | 25.3 | 40.4 | 45.2 | 38.0 |
| + **DEVA** (Full Fine-tuning) | 48.9 | 47.3 | 42.3 | 49.5 | 53.5 | 48.9 |
| + **DEVA** (Rank = 16, Attach: LLM) | 45.2 | 44.0 | 38.2 | 46.2 | 50.3 | 45.3 |
| + **DEVA** (Rank = 32, Attach: LLM) | 45.7 | 44.5 | 38.8 | 46.9 | 50.9 | 46.0 |
| + **DEVA** (Rank = 64, Attach: LLM) | 46.1 | 45.2 | 39.5 | 47.6 | 51.5 | 46.9 |
| + **DEVA** (Rank = 16, Attach: VE) | 43.7 | 42.2 | 36.3 | 44.0 | 48.2 | 43.2 |
| + **DEVA** (Rank = 32, Attach: VE) | 44.2 | 43.0 | 37.0 | 44.8 | 48.9 | 43.9 |
| + **DEVA** (Rank = 64, Attach: VE) | 44.9 | 43.8 | 38.0 | 45.4 | 49.6 | 44.6 |
| + **DEVA** (Rank = 16, Attach: VE + LLM) | 46.5 | 45.9 | 40.5 | 48.0 | 52.0 | 47.1 |
| + **DEVA** (Rank = 32, Attach: VE + LLM) | 47.9 | 46.8 | 41.2 | 48.8 | 52.9 | 47.8 |
| + **DEVA** (Rank = 64, Attach: VE + LLM) | 48.5 | 47.1 | 41.9 | 49.1 | 53.4 | 48.5 |

## 6. Effect on other models

In this subsection, we test whether our method is applicable to other models: GLM-Edge [8] and LLAVA [16] in Table 5. We observe that our framework DEVA produces significant improvement in IoU over Visual-RFT and also surpasses the IoU of GSPO. However, the gap between GSPO and our proposed DEVA framework is diminished for LLAVA1.5-7B. Overall, we see DEVA is more effective for smaller models. This suggests that our framework can be very effective for small-scale devices to be deployed on edge devices.

Table 5. **Reasoning Grounding Results on LISA [13]**. using the GLM-Edge model [8]. and LLAVA1.5 [16]

| Model | $mIoU_{test}$ | $mIoU_{val}$ | $gIoU_{test}$ | $mIoU_{test}$ | $mIoU_{val}$ | $gIoU_{test}$ |
|---|---|---|---|---|---|---|
| GLM-Edge-V-2B \| LLAVA1.5-7B | 24.4 | 27.5 | 22.5 | 38.9 | 42.3 | 35.4 |
| + SFT | 26.8 | 27.2 | 22.6 | 36.2 | 41.1 | 34.3 |
| + SFT-CoT | 28.6 | 31.2 | 25.6 | 38.3 | 42.7 | 36.2 |
| + PPO [24] | 31.1 | 34.4 | 30.4 | 39.2 | 43.4 | 37.3 |
| + PAPO [27] | 35.4 | 38.7 | 32.9 | 42.0 | 45.3 | 41.0 |
| + DAPO [29] | 36.9 | 40.0 | 34.4 | 42.2 | 45.3 | 41.0 |
| + Dr GRPO [18] | 35.4 | 38.7 | 33.5 | 41.5 | 45.7 | 41.2 |
| + BNPO [28] | 35.3 | 38.6 | 34.2 | 41.7 | 46.1 | 41.4 |
| + GRPO-CARE [2] | 36.9 | 40.0 | 33.3 | 42.3 | 46.3 | 41.9 |
| + CPPO [15] | 37.6 | 40.8 | 33.4 | 42.8 | 46.4 | 42.2 |
| + GMPO [30] | 36.3 | 40.5 | 32.9 | 41.8 | 46.2 | 41.8 |
| + GSPO [31] | 38.5 | 42.0 | 34.3 | 43.2 | 47.1 | 43.3 |
| + ViRFT [19] | 34.8 | 31.8 | 31.6 | 42.0 | 44.3 | 41.0 |
| + **DEVA** (**Div.**) | 41.1 | 42.2 | 36.6 | 43.2 | 47.3 | 44.2 |
| + **DEVA** (**Div.** + **Explor.**) | 42.2 | 43.3 | 37.3 | 44.3 | 48.8 | 45.0 |
| + **DEVA** (**Div.** + **Explor.** + Align. **Vol.**) | <u>44.2</u> | <u>44.4</u> | <u>38.5</u> | <u>45.2</u> | <u>50.0</u> | <u>45.4</u> |
| + **DEVA** (**Div.** + **Explor.** + Align. **Vol.** + **Agg.**) | **46.4** | **44.8** | **39.5** | **46.6** | **50.7** | **46.1** |

We also apply DEVA to other models: Qwen2.5-VL-3B-Instruct model [1] and Qwen3-VL-8B-Instruct [16] in Table 6 and observe similar increase in performance. This shows that our method is also generalizable to models with increasing

capacity as well.

Table 6. **Reasoning Grounding Results on LISA [13]**. using the Qwen2.5-VL-3B-Instruct model [1] and Qwen3-VL-8B-Instruct [16]

| Model | $\text{mIoU}_{\text{test}}$ | $\text{mIoU}_{\text{val}}$ | $\text{gIoU}_{\text{test}}$ | $\text{mIoU}_{\text{test}}$ | $\text{mIoU}_{\text{val}}$ | $\text{gIoU}_{\text{test}}$ |
|---|---|---|---|---|---|---|
| Qwen2.5-VL-3B-Instruct \| Qwen3-VL-8B-Instruct | 27.8 | 30.6 | 26.0 | 42.6 | 45.8 | 38.9 |
| + SFT | 30.4 | 30.8 | 26.4 | 39.8 | 44.9 | 37.8 |
| + SFT-CoT | 32.3 | 34.6 | 29.0 | 41.9 | 46.6 | 39.8 |
| + PPO [24] | 34.8 | 37.9 | 33.8 | 42.9 | 47.3 | 40.9 |
| + PAPO [27] | 38.9 | 42.3 | 36.6 | 45.8 | 49.0 | 44.6 |
| + DAPO [29] | 40.6 | 43.6 | 37.9 | 46.0 | 49.1 | 44.8 |
| + Dr GRPO [18] | 38.9 | 42.4 | 36.9 | 45.1 | 49.4 | 44.9 |
| + BNPO [28] | 38.8 | 42.2 | 37.6 | 45.4 | 49.9 | 45.2 |
| + GRPO-CARE [2] | 40.6 | 43.7 | 36.8 | 46.0 | 50.0 | 45.6 |
| + CPPO [15] | 41.3 | 44.4 | 36.9 | 46.6 | 50.3 | 46.0 |
| + GMPO [30] | 40.0 | 44.1 | 36.4 | 45.6 | 50.0 | 45.6 |
| + GSPO [31] | 42.3 | 45.7 | 37.8 | 47.0 | 50.9 | 47.0 |
| + ViRFT [19] | 38.5 | 35.4 | 35.1 | 45.7 | 48.0 | 44.7 |
| + **D**EVA (**D**iv.) | 44.8 | 45.9 | 40.1 | 46.9 | 51.0 | 47.8 |
| + **DE**VA (**D**iv. + **E**xplor.) | 45.9 | 47.0 | 40.8 | 48.0 | 52.5 | 48.6 |
| + **DEV**A (**D**iv. + **E**xplor. + Align. **V**ol.) | <u>47.9</u> | <u>48.1</u> | <u>42.0</u> | <u>48.9</u> | <u>53.7</u> | <u>49.0</u> |
| + **DEVA** (**D**iv. + **E**xplor. + Align. **V**ol. + **A**gg.) | **50.2** | **48.6** | **43.0** | **50.3** | **54.4** | **49.7** |

## 7. Mask Computation for Mapping Images for Alignment Volume

Our goal is to obtain the binary mask $m$ in the image for computing alignment reward defined in Eq. 5. For computing the binary mask $m$, we need to do a forward pass of the image and text query through the multi-modal large language model to obtain attention scores and backtrack them to the image to obtain relevant patches. The details of obtaining relevant patches are described below.

**Attention-to-patch mask.** We consider that $i \in \mathbb{R}^{H\times W\times 3}$ is resized by the image processor to $i' \in \mathbb{R}^{H'\times W'\times 3}$. If we have patch size $p = 14$, the visual encoder yields a grid of patches of size $H_p = H'/p$, $W_p = W'/p$, and $N_v = H_pW_p$ visual tokens. In the multimodal sequence, visual tokens are arranged continuously as $\mathcal{V} = \{\, tok \mid tok_{\text{vs}} < tok < tok_{\text{ve}} \,\}$ between special tokens at positions $tok_{\text{vs}}$ and $tok_{\text{ve}}$, respectively.

**Decoder attentions over visual tokens.** At decoding step $t$ (when predicting text token $o_t$), layer $\ell \in \{1, \dots, L\}$ and head $h \in \{1, \dots, H\}$ produces self-attention matrix $A^{(\ell,h,t)} \in \mathbb{R}^{T_t\times T_t}$, whose row $t$ is distribution over source positions $i \in \{1, \dots, T_t\}$. We define an aggregated score such that

$$s_i^{(t)} \;=\; \sum_{\ell=1}^{L}\sum_{h=1}^{H} w_\ell\, u_h\, A_{t,i}^{(\ell,h,t)} \qquad (i \in \mathcal{V}), \tag{1}$$

with nonnegative weights $w_\ell, u_h$ such that $\sum_{\ell=1}^{L} w_\ell = 1$ and $\sum_{h=1}^{H} u_h = 1$.

We apply a min–max normalization over visual positions:

$$\tilde{s}_i^{(t)} \;=\; \frac{s_i^{(t)} - \min_{j\in\mathcal{V}} s_j^{(t)}}{\max_{j\in\mathcal{V}} s_j^{(t)} - \min_{j\in\mathcal{V}} s_j^{(t)} + \varepsilon} \;\in\; [0,1]. \tag{2}$$

**Aggregating multiple answer tokens.** We consider the case when the mask considers multiple output tokens. In that case, we let $\mathcal{T}$ be the indices and $v_t \geq 0$ with $\sum_{t\in\mathcal{T}} v_t = 1$. We define

$$\tilde{s}_i \;=\; \sum_{t\in\mathcal{T}} v_t\, \tilde{s}_i^{(t)}. \tag{3}$$

**Mapping visual tokens to the patch grid.** Index visual tokens locally as $k \in \{1,\dots,N_v\}$ (in order within $\mathcal{V}$). Map $k$ to patch coordinates $(r,c)$ via

$$r \;=\; 1 + \left\lfloor \frac{k-1}{W_p} \right\rfloor, \qquad c \;=\; 1 + \big((k-1) \bmod W_p\big). \tag{4}$$

Let $i(k)$ denote the absolute sequence index corresponding to the $k$-th visual token. The patch-level score map $S \in [0,1]^{H_p\times W_p}$ is

$$S_{r,c} \;=\; \tilde{s}_{\,i(k)}. \tag{5}$$

**Upsampling and binarization.** Let $\mathcal{U}_p$ be bilinear upsampling by factor $p$. The soft mask over $I'$ is

$$M \;=\; \mathcal{U}_p(S) \;\in\; [0,1]^{H'\times W'}. \tag{6}$$

A binary mask at threshold $\tau \in (0,1)$ is given by

$$m(x,y) \;=\; \mathbf{1}[\,M(x,y) \geq \tau\,]. \tag{7}$$

The threshold $\tau$ is given as 0.5.

## 8. Justification for Alignment Volume

We can consider the case with three modalities: $f_i$ (input), $f_q$ (query), and $f_o$ (output). The Gram Matrix is equal to:

$$G = \begin{bmatrix} f_i \cdot f_i & f_i \cdot f_q & f_i \cdot f_o \\ f_q \cdot f_i & f_q \cdot f_q & f_q \cdot f_o \\ f_o \cdot f_i & f_o \cdot f_q & f_o \cdot f_o \end{bmatrix}$$

We can compute the determinant of the matrix $G$:

$$\begin{aligned} \det(G) = \langle f_i, f_i\rangle \cdot (\langle f_q, f_q\rangle \cdot \langle f_o, f_o\rangle - \langle f_q, f_o\rangle \cdot \langle f_o, f_q\rangle) & \quad (8)\\ - \langle f_i, f_q\rangle \cdot (\langle f_q, f_i\rangle \cdot \langle f_o, f_o\rangle - \langle f_q, f_o\rangle \cdot \langle f_o, f_i\rangle) & \quad (9)\\ + \langle f_i, f_o\rangle \cdot (\langle f_q, f_i\rangle \cdot \langle f_o, f_q\rangle - \langle f_q, f_q\rangle \cdot \langle f_o, f_i\rangle) & \quad (10) \end{aligned}$$

It is to be noted that the $f_i, f_q, f_o$ embeddings are normalized to unit norm and hence $f_i f_i = f_q f_q = f_o f_o = 1$.
Using that, we obtain the determinant of the Gram matrix $G$ such that

$$\begin{aligned} \det(G) &= 1 \cdot (1 - f_o f_q^2) - f_i f_q \cdot (f_q f_i - f_q f_o \cdot f_o f_i) + f_i f_o \cdot (f_q f_i \cdot f_o f_q - f_o f_i) && (11)\\ &= 1 - f_o f_q^2 - f_i f_q^2 + f_i f_q \cdot f_q f_o \cdot f_o f_i + f_i f_o \cdot f_q f_i \cdot f_o f_q - f_i f_o^2 && (12)\\ &= 1 - f_o f_q^2 - f_i f_q^2 - f_i f_o^2 + 2 \cdot f_i f_q \cdot f_q f_o \cdot f_o f_i && (13) \end{aligned}$$

As can be seen from the equations above, there are cross product computations which produces an alignment of all the modalities together. The other alternative is pair-wise cosine similarity that computes similarities between pairs of modalities which might not be optimal as described in the introduction. This theoretical justification of using volume for alignment has also been considered in [4].

## 9. Different Reward Aggregation Techniques

In this method, we consider different reward aggregation techniques like arithmetic mean, geometric mean, harmonic mean and neural network, etc. Let us consider that we have three types of rewards: format reward $r_{form}$, task reward $r_{task}$ and volume reward $r_v$. In that case, we consider the following types of aggregation.

**Arithmetic Sum:** For the scaled arithmetic mean, we consider the aggregated reward $r = (r_{form} + r_{task} + r_v)$.

**Geometric Mean:** For the scaled geometric mean, we consider the aggregated reward $r = 3(r_{form} r_{task} r_v)^{1/3}$

**Harmonic Mean:** For the scaled harmonic mean, we consider the aggregated reward $r = 9/((1/r_{form}) + (1/r_{task}) + (1/r_v))$

**Neural Network:** For the neural network $\Phi(\cdot)$, we consider the following formulation for prediction. It takes in the three reward scalars $r_{form}$, $r_{task}$ and $r_v$ and produces an aggregated reward $r$ such that $r = \Phi(r_{form}, r_{task}, r_v)$. When we use this neural network, it is a multi-stage training procedure:

- **Stage 1:** We train the policy model with the harmonic reward using the GRPO training objective for the same epoch numbers as standard fine-tuning.
- **Stage 2:** With the same GRPO training objective, we freeze the policy model and train the neural network based predictor that takes in three reward scalars to produce the desired reward. This training is done for half the number of epochs as standard fine-tuning.
- **Stage 3:** During the final stage, we freeze the neural network based predictor and fine-tune the policy model for the same number of epochs as standard fine-tuning.

The neural network architecture is two-layered with input size of 3, hidden state size of 2 and output size as 1.

## 10. Additional Hyperparameter Studies

In Table 7, we observe how the mIoU varies for different variations. With respect to the entropy divergence loss defined in Eq. 4, we consider the following variations: (a) Partition 2: When mean squared error is computed separately for 2 partitions of the tokens of the reference model and policy model. (b) Partition 3: When mean squared error is computed separately for 3 partitions of the tokens of the reference model and the policy model. (c) OT: We consider the optimal transport distance [7] between the two entropy vectors obtained from the reference model and the policy model. The cost matrix is computed such that each element is the cosine distance between CLIP [23] embedding of the two words.

Furthermore, we consider feature extractors defined in Eq. 5. This includes CLIP [23] and SigLip2 [25]. Overall, we observe that a model with larger capacity produces better performance. However, all model variants produce poorer performance compared to the default version of DEVA.

In our framework, we consider the GFlowNet training objective for improving diversity of rewards. As an alternative, we could also consider the Pass@k metric as a reward for improved reward diversity and improved recognition performance. This has been explored in [3], where they use the metric for improved diversity of generation in LLMs. However, as observed in Table 8, using Pass@k as an additional reward instead of GFlowNet, produces a drop of about 2 pts.

In Table 7, we vary $\gamma$ from the default value of 0.5. This $\gamma$ is used for the diversity loss defined in Eq. 3. As we can see in the results of Table 7, alternative values of $\gamma$ produce drop in performance compared to the optimal value $\gamma = 0.5$.

## 11. Additional Computational Overhead

The results of additional computational overhead is shown in Table 8. The Peak RAM memory is measured on a per GPU basis.

## 12. Correlation of alignment volume with improved IoU

We also show results in Figure 1 that highlight how effective the alignment volume reward is in improving IoU. Specifically, for a single test sample of the LISA reasoning dataset, we obtain IoU with two checkpoints: (a) where Diversity and Exploration loss is applied. (b) where Diversity loss, Exploration loss and the Alignment Reward is applied. We obtain the difference between these two IoUs and plot it as the Y-axis. For the X-axis, we obtain the Alignment volume for the test sample inputs with respect to the predictions. This is plotted for all the test samples. The method is repeated across multiple RL algorithms: (a) ViRFT (b) PAPO (c) DAPO (d) GSPO. Across all these algorithms, we obtain a negative correlation between the improved IoU and the volume. This suggests that including the alignment volume based reward is infact effective in improving grounding performance.

Table 7. **Reasoning Grounding Results on LISA [13]**. Selected metrics are shown for different model variations.

| Model | $\text{mIoU}_{\text{test}}$ | $\text{mIoU}_{\text{test}}$ |
|---|---|---|
| Qwen2-VL-2B \| 7B [26] | 26.9 | 40.4 |
| + **DEVA** (Default) | 48.9 | 49.5 |
| + **DEVA** (Explor. Loss: Partition=2) | 48.0 | 48.7 |
| + **DEVA** (Explor. Loss: Partition=3) | 46.5 | 47.6 |
| + **DEVA** (Explor. Loss: OT) | 48.5 | 48.9 |
| + **DEVA** (Embed: CLIP B-16) | 46.3 | 47.1 |
| + **DEVA** (Embed: CLIP L-14) | 47.5 | 48.9 |
| + **DEVA** (Embed: SigLip2 B-16) | 46.9 | 47.8 |
| + **DEVA** (Embed: SigLip2 L-16) | 47.8 | 49.0 |
| + **DEVA** (Pass@k [3]) | 46.8 | 47.4 |
| + **DEVA** ($\gamma = 0.25$) | 46.3 | 47.1 |
| + **DEVA** ($\gamma = 0.75$) | 46.5 | 47.0 |

Table 8. **Computational Overhead on LISA [13]**. Selected metrics are shown for different model variations.

| Model | Training Time (mins) | Peak RAM Memory (GB) |
|---|---|---|
| Qwen2-VL-2B [26] + ViRFT [19] | 212 | 58.2 |
| + **D**EVA (**D**iv.) | 255 | 63.2 |
| + **DE**VA (**D**iv. + **E**xplor.) | 262 | 65.6 |
| + **DEV**A (**D**iv. + **E**xplor. + Align. **V**ol.) | 325 | 75.4 |
| + **DEVA** (**D**iv. + **E**xplor. + Align. **V**ol. + **A**gg.) | 331 | 77.1 |

## 13. Additional Studies

We experimented on the MMLU [10] and GSM8K [5] benchmark using pure LLM backbone of Qwen2.5-14B and show that DEVA (without the multimodal volume reward) can still produce improved performance as shown in Table 9 below.

Table 9. Experiments on LLM backbone.

| **Dataset** | **Baseline** | **+Div.** | **+Div. +Explor.** | **+Div. +Explor. + Agg.** |
|---|---|---|---|---|
| MMLU | 79.70 | 81.30 | 82.20 | 83.30 |
| GSM8K | 90.20 | 91.0 | 91.80 | 92.50 |

Global entropic divergence balances exploration and stability, constraining information without token-level overfitting for vision tasks. We keep format rewards and monitor token-level KL. To fully address the reviewer's concern on length sensitivity and localized degeneration, we add: (i) **Test 1:** mIoU on LISA reasoning for different decoding lengths 64/128/256/512 (ii) **Test 2:** Nonsense span rate (NSR): % of outputs with any window whose perplexity exceeds the reference's 95-th percentile for different decoding lengths 64/128/256/512. The results in table 10 suggest that both metrics are robust across different lengths. We would also add effect on varying decoding temperature and entropy spike rates over different training steps.

The variance results for Table 1 and 2 on 5 seeds as shown below in Table 11 and 12, remain stable. In figure 2, on LISA dataset, diversity plots show higher avg. std. of rewards when adding GFlowNet loss. Also, per-token entropy increases because of global entropy regularization and exploration.

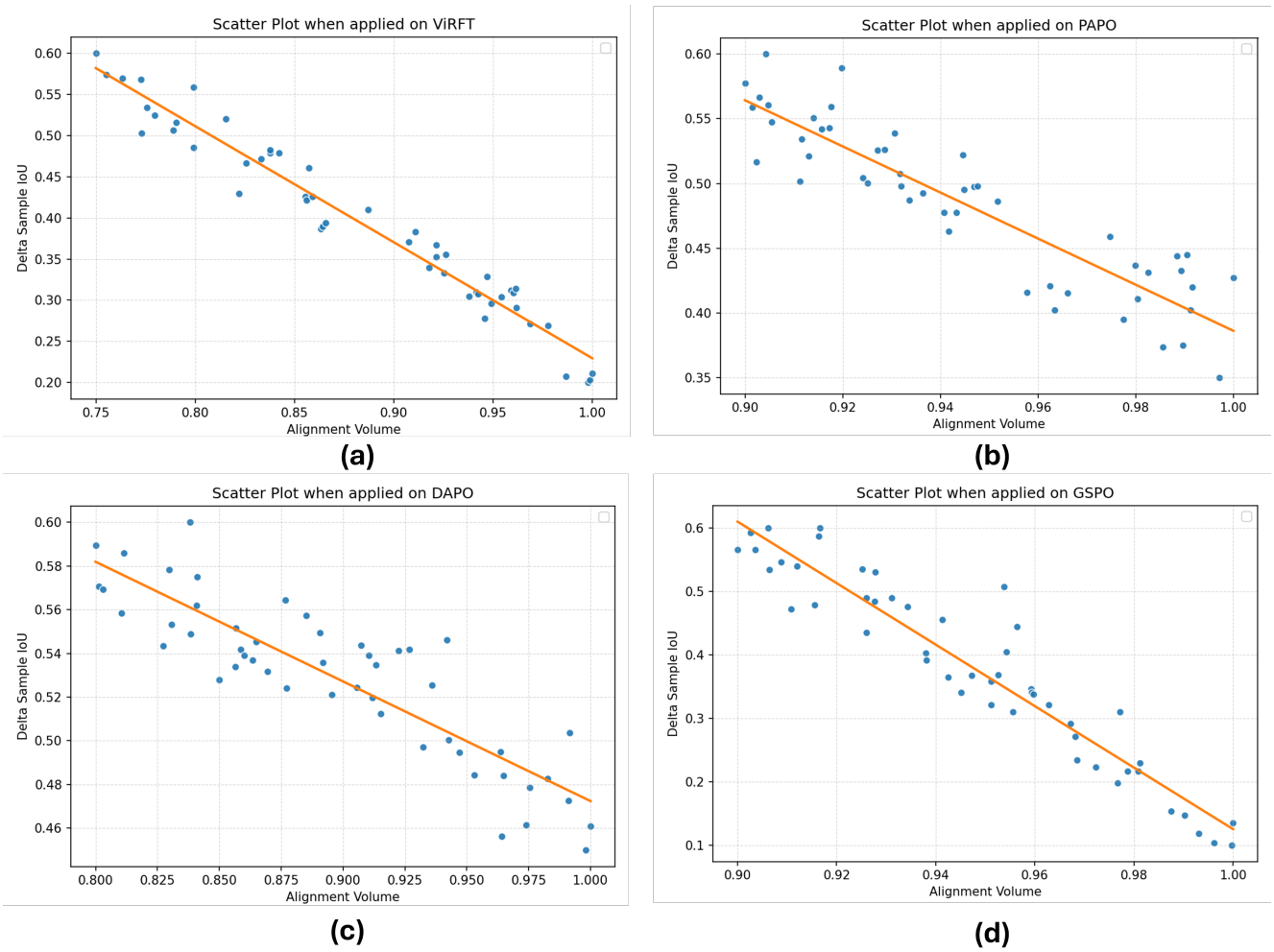


Figure 1. Plot of change in IoU for test sample of LISA dataset versus the alignment volume observed for the test sample. The change in IoU is between X+**DE**VA and X+**DEV**A. Here X is one of the RL algorithms in (a) ViRFT (b) PAPO (c) DAPO (d) GSPO.

Table 10. Robustness Tests.

| **Setup** | **Baseline (64)** | **128** | **256** | **512** |
|---|---|---|---|---|
| mIoU | 79.70 | 81.30 | 82.20 | 83.30 |
| NSR (%) | 10.12 | 9.86 | 10.22 | 10.01 |

Table 11. Variance results for table 1 in the main paper.

| **Model** | 1-shot | 2-shot | 4-shot | 8-shot | 16-shot | 4-shot |
|---|---|---|---|---|---|---|
| + **DE**VA (**D**iv.) | 0.01 (0.03) | 0.01 (0.04) | 0.02 (0.03) | 0.05 (0.04) | 0.02 (0.03) | 0.02 |
| + **DE**VA (**D**iv. + **E**xplor.) | 0.02 (0.01) | 0.03 (0.02) | 0.03 (0.01) | 0.02 (0.02) | 0.01 (0.04) | 0.02 |

Table 12. Variance results for table 2 in the main paper.

| **Model** | $\text{mIoU}_{\text{test}}$ | $\text{mIoU}_{\text{val}}$ | $\text{gIoU}_{\text{test}}$ | $\text{mIoU}_{\text{test}}$ | $\text{mIoU}_{\text{val}}$ | $\text{gIoU}_{\text{test}}$ |
|---|---|---|---|---|---|---|
| + **DEVA** (**Div.**) | 0.03 | 0.06 | 0.08 | 0.01 | 0.02 | 0.01 |
| + **DEVA** (**Div.** + **Explor.**) | 0.01 | 0.02 | 0.02 | 0.01 | 0.04 | 0.05 |

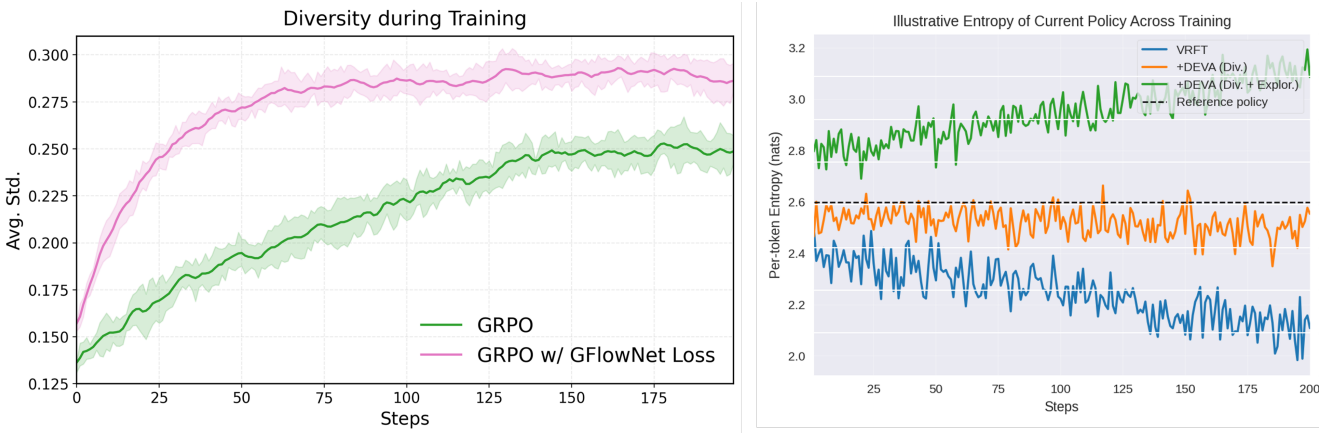


Figure 2. Training metrics over epochs, showing the evolution of model performance during optimization.

## 14. Additional Visualization

In this section, we show additional visualization for some samples on the fine-grained classification and LISA reasoning grounding datasets. We visualize the heatmaps as well as the bounding boxes for the LISA grounding datasets. From the results, we see that DEVA produces better localization capabilities compared to other methods, where the heatmap focuses more around the objects of interest. Furthermore, we observe that with DEVA, introducing additional components of the framework progressively produces better heatmap localization and focus.

## References


[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[2] Yi Chen, Yuying Ge, Rui Wang, Yixiao Ge, Junhao Cheng, Ying Shan, and Xihui Liu. Grpo-care: Consistency-aware reinforcement learning for multimodal reasoning. *arXiv preprint arXiv:2506.16141*, 2025.

[3] Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@ k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*, 2025.

[4] Giordano Cicchetti, Eleonora Grassucci, Luigi Sigillo, and Danilo Comminiello. Gramian multimodal representation learning and alignment. *arXiv preprint arXiv:2412.11959*, 2024.

[5] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[6] Federico Cocchi, Nicholas Moratelli, Davide Caffagni, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. Llava-more: A comparative study of llms and visual backbones for enhanced visual instruction tuning. *arXiv preprint arXiv:2503.15621*, 2025.

[7] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016.

[8] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

[9] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019.


Figure 3. Visualization of the Chihuahua as part of the Pets37 dataset.

[10] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[12] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV workshops*, 2013.

[13] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2024.

[14] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[15] Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. Cppo: Accelerating the training of group relative policy optimization-based reasoning models. *arXiv preprint arXiv:2503.22342*, 2025.

[16] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[17] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024.

[18] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

[19] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*, 2025.

[20] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

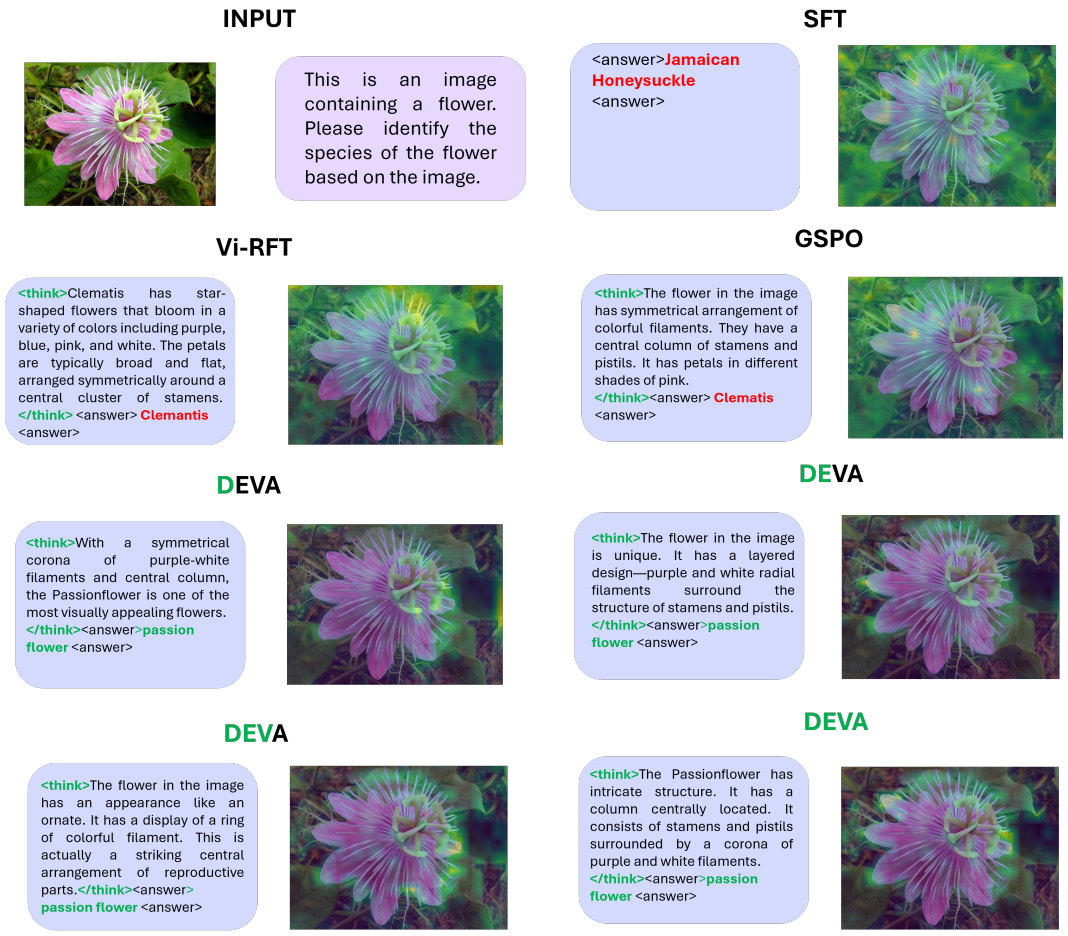


Figure 4. Visualization of the Passion Flower as part of the Flower102 dataset.


[21] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

[22] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012.

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

[24] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[25] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[26] 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*, 2024.

[27] Zhenhailong Wang, Xuehang Guo, Sofia Stoica, Haiyang Xu, Hongru Wang, Hyeonjeong Ha, Xiusi Chen, Yangyi Chen, Ming Yan, Fei Huang, et al. Perception-aware policy optimization for multimodal reasoning. *arXiv preprint arXiv:2507.06448*, 2025.

[28] Changyi Xiao, Mengdi Zhang, and Yixin Cao. Bnpo: Beta normalization policy optimization. *arXiv preprint arXiv:2506.02864*, 2025.

[29] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale, 2025. *URL https://arxiv. org/abs/2503.14476*, 2025.

[30] Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025.

[31] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

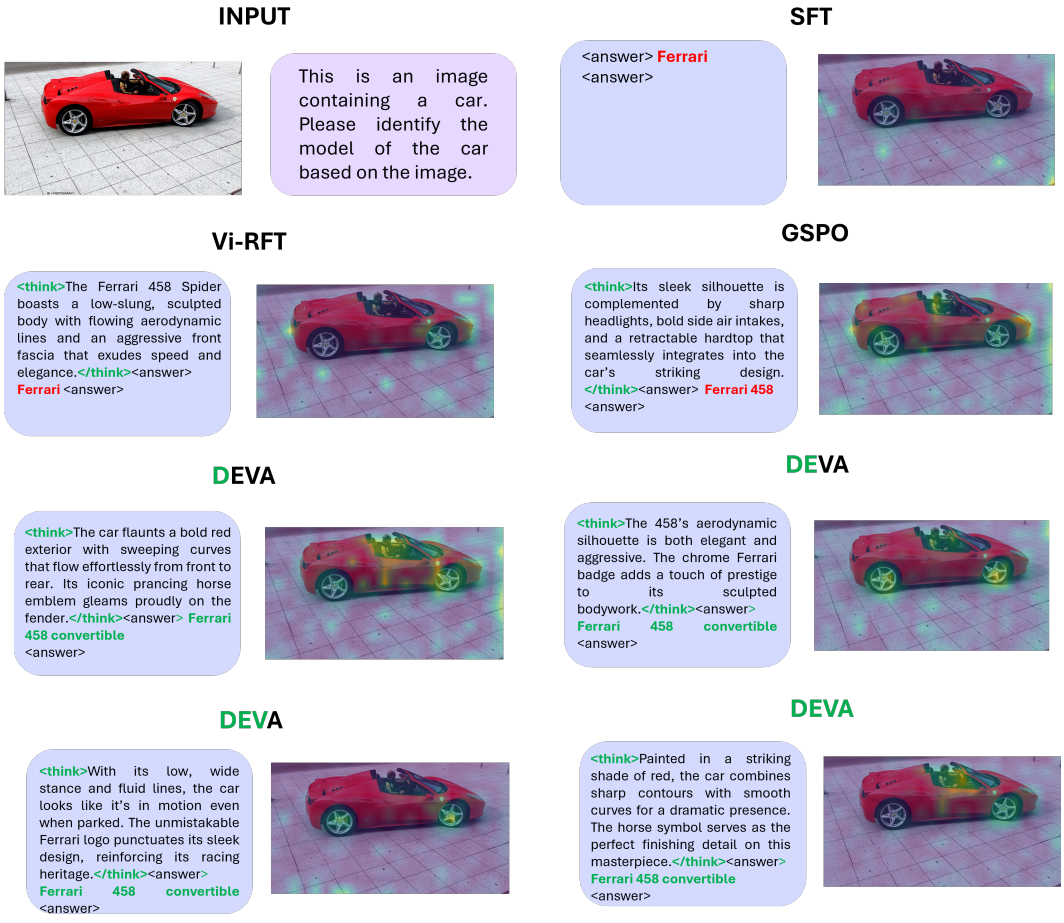


Figure 5. Visualization of the Ferrari car as part of the Stanford Cars dataset.

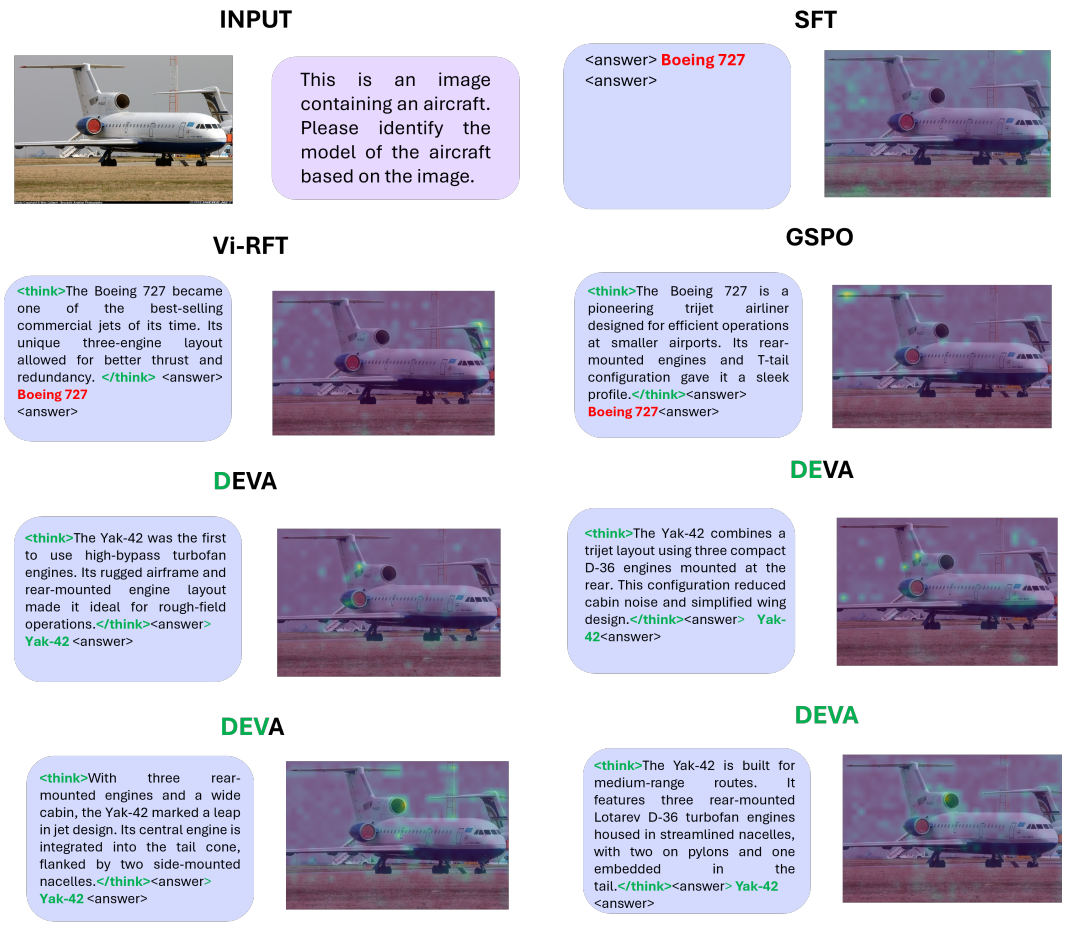


Figure 6. Visualization of the Airplane as part of the FGVC dataset.

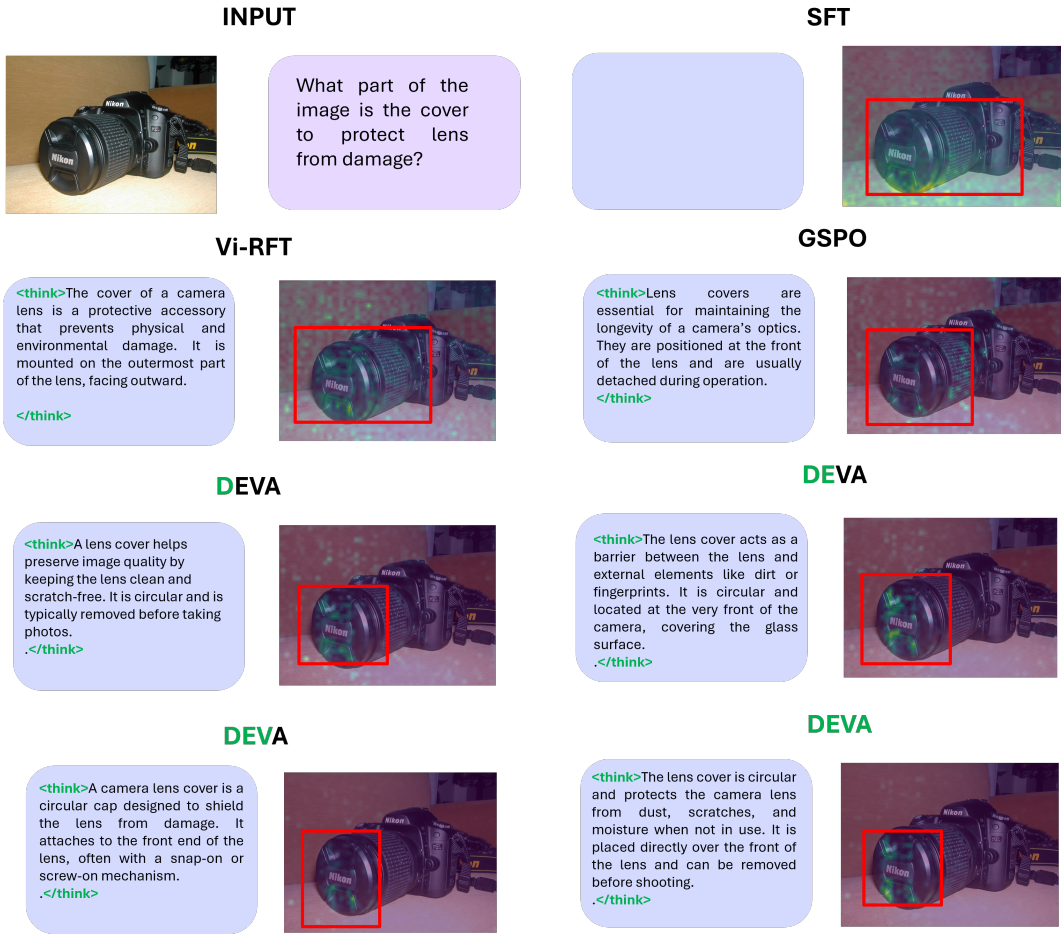


Figure 7. Visualization of the Camera as part of the LISA reasoning grounding dataset.


Figure 8. Visualization of the Chopstick as part of the LISA reasoning grounding dataset.

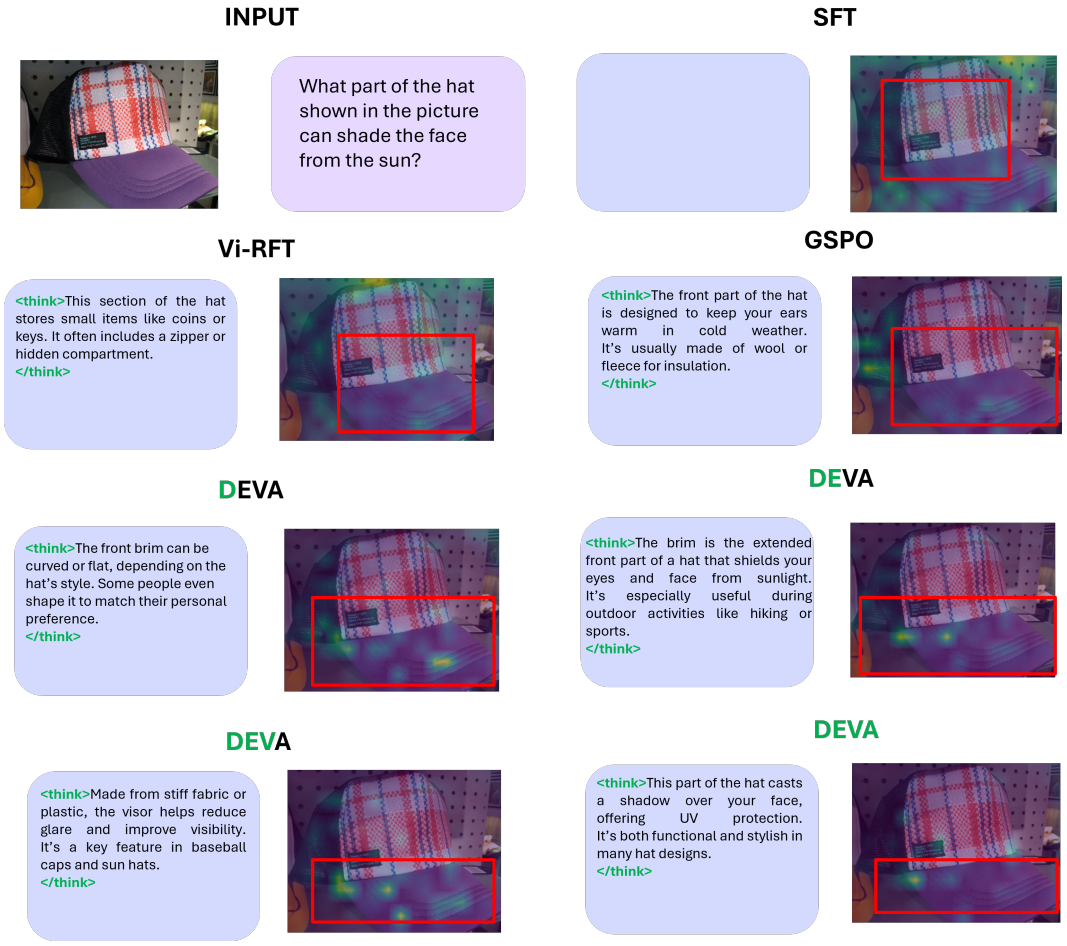


Figure 9. Visualization of the Hat as part of the LISA reasoning grounding dataset.

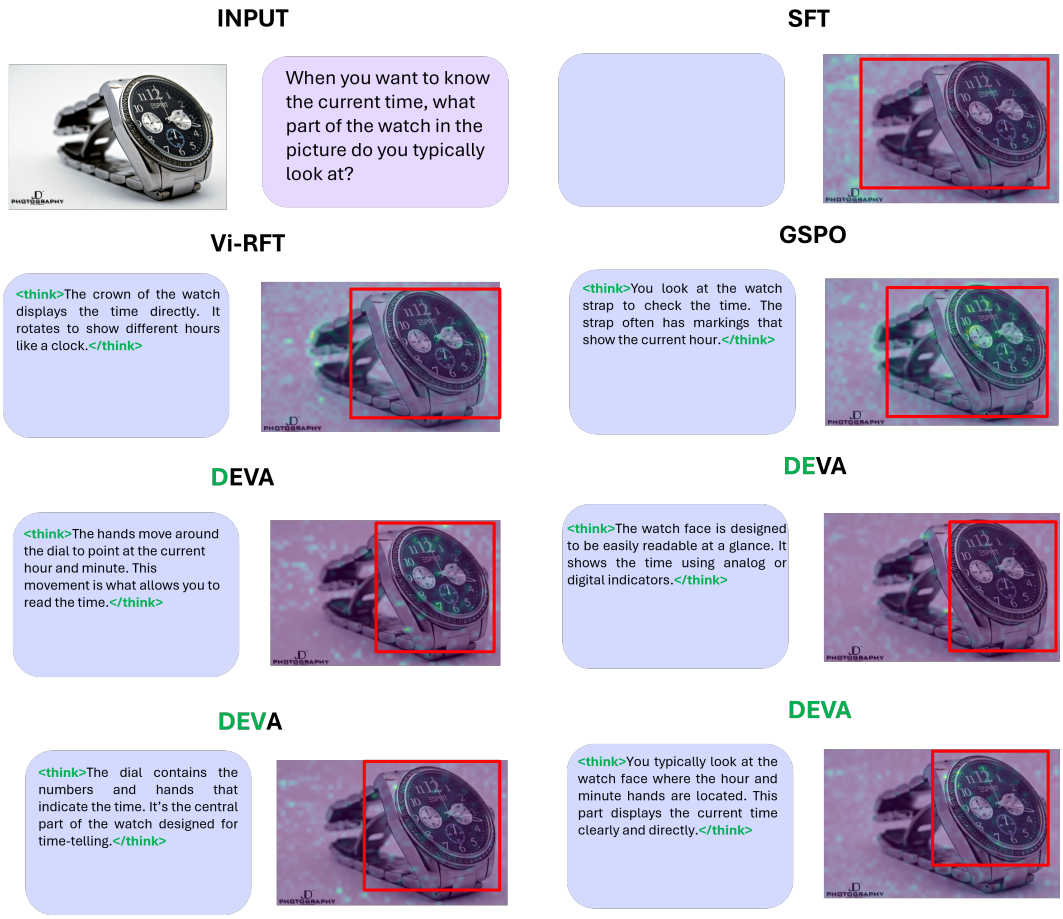


Figure 10. Visualization of the Watch as part of the LISA reasoning grounding dataset.