Same or Not? Enhancing Visual Perception in Vision-Language Models

Supplementary Material

Contents

A. Additional details on FGVQA

We provide additional details on FGVQA, our new benchmark suite for evaluating fine-grained VQA. We describe the datasets included in FGVQA in Sec. A.1. We illustrate additional examples in Fig. 7, and describe all prompts used in Sec. A.2. We report exact match accuracy for all benchmarks on both *pair* and *multi* queries.

A.1. Evaluation Benchmarks

We describe the provenance of each benchmark in FGVQA below.

TWIN-Eval is the evaluation set of TWIN. It is collected identically to TWIN (Sec. 3), but features distinct instances and images from TWIN.

ILIAS [33] is a large-scale test dataset of instance-level image retrieval. It predominantly features images of retail products taken in various contexts, backgrounds, and lighting. For both *pair* and *multi* queries, we source images from the `core_db` split, using the `__key__` field to determine instance identity.

	MEAN	TWIN-Eval	ILIAS	LANDMARKS	MET	CUB	INQUIRE
Human	92.9	90.0	100.0	82.5	93.8	92.5	98.8

Table 7. **Human Evaluations on FGVQA.**

Google Landmarks v2 (LANDMARKS) [97] is a landmark recognition dataset featuring human-made and natural landmarks. The dataset has been used in both classification and retrieval settings. To ensure sufficient images per landmark to for *multi* queries, we source images from the `train` split and use the `label` field to determine landmark identity.

MET [107] is an image retrieval dataset featuring artwork from the Metropolitan Museum of Art in New York. The dataset features images of the same art piece or sculpture from varying viewpoints, emphasizing multi-view consistency in retrieval. We use the `mini_met` set to construct both *pair* and *multi* queries.

CUB [91] is a fine-grained classification dataset that focuses on identifying bird species from images. We use the `test` split for all queries and create *pair* and *multi* queries based on the specie of bird.

Inquire [88] is a benchmark for natural world image retrieval, featuring images of animal and plant species sourced from iNaturalist [87]. We source images from the `validation` split and use the `inat24_species_name` field to determine instance identity for both *pair* and *multi* queries.

A.2. FGVQA Prompts

We include all prompts used on all datasets for both *pair* and *multi* queries in Figs. 12 to 17. All prompts are identical in format, except for the domain-specific instance names (*e.g.* bird/object/artwork/species). Additionally, for benchmarks featuring object instances (TWIN-Eval, ILIAS [33]), we found it necessary to provide the definition of an object instance as models were incorrectly labeling different objects of the same-category (*e.g.* two earbuds of different colors) as the same.

A.3. FGVQA Human Evaluation

We establish human performance on FGVQA in Tab. 7. We task four human annotators with repsonding to a subset of 20 queries per benchmark in FGVQA. We report total accuracy (%) on each dataset, averaged across evaluator. We find that human evaluators significantly outperform open-source VLMs, suggesting ample future work is needed in fine-grained VQA.

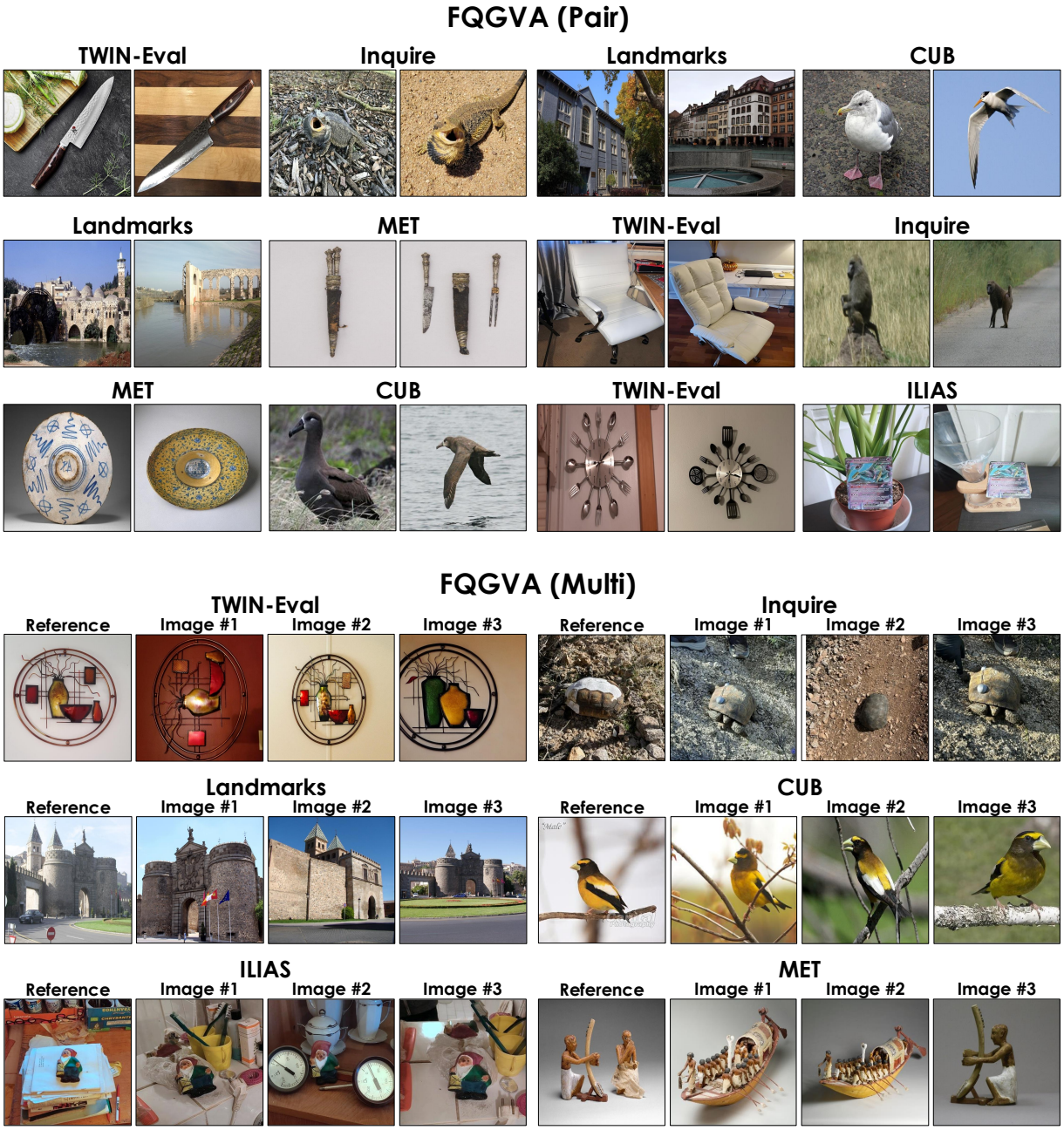

Figure 7. **Additional examples from FGVQA.** We show examples of images in *pair* and *multi* queries. FGVQA spans six datasets across a wide range of domains, creating a challenging suite for assessing cross-domain fine-grained VQA in VLMs.

B. Additional details on TWIN

We provide further details on the collection of TWIN, including instance sourcing, de-duplication, hard negative pair assignment, and details on generating synthetic negatives.

B.1. Instance Sourcing

We used the Amazon Reviews dataset [24] to extract a large set of product listings, which would serve as candidate objects, and associated noisy review images. We categorized these objects into supercategories based on their associated category in the Amazon store (*e.g.* "Electronics") and further into categories based on the product title (*e.g.* "Speaker"). We aimed to maximize diversity in

our instances and thus selected 36 categories to draw from. We subsequently tasked human annotators with removing products that either lacked visual consistency (*e.g.* the same product being sold in multiple colors) or did not adhere to the object category (*e.g.* "earbud replacement tips" instead of "earbuds").

B.2. De-Duplication

To ensure the correctness of the pairwise labels in TWIN, we must ensure that object instances are *unique*. If object instances are not unique, then two of the same objects could incorrectly be listed as negative pairs. However, on Amazon duplicate listings are common. We therefore merged duplicate objects into the same instance. To this end, we used a pre-trained CLIP model to compare the images of each product with the images of all other products via cosine similarity. We flagged the k images with the highest similarity as potential duplicates, setting k equal to the number of images for the product. We then tasked human annotators with manually inspecting the flagged images. If a duplicate was found, we merged the two products. The resulting object instances after de-duplication are used to produce the 561K instances of TWIN.

B.3. Hard Negative Pair Assignment

Once we have collected a set of object instances, we construct hard negative pairs – pairs of distinct objects that appear similar. We collect these hard negatives with the help of human annotators. First, using CLIP [67], for each object instance, we compute pairwise cosine similarities between the image embeddings of that instance and all other instances. We use these similarities to shortlist k visually similar candidates for each instance, with k being a random number between $1 - 2\times$ the number of positive images of that instance. Human annotators then select the final pairs, omitting pairs they deem to be too easy.

B.4. Additional Details on Generated Negatives

We augment the set of negative pairs with additional synthetically generated images. Augmenting with generated negatives allows us to produce additional negative pairs without extra data collection. We follow the procedure outlined in Dreambooth [70]. For each object instance, we sample 3 to 5 images for each instance to serve as grounding images. We set the initializer tokens to the generic category associated with the instance (*e.g.* "vase", "headphones"). Our prompts to Dreambooth are: ["an image of {category}", "similar to {category}", "a picture of{category}, "show me a {category}, "here is a {category}"] replacing {category} with the category of the instance. Finally, we task human annotators with validating the generated samples. Annotators are asked to remove generations that either do not appear similar to the instance or are indistinguishable from the instance – yielding a set of hard negative pairs.

We show additional examples of synthetic negative pairs in Fig. 8. For each pair of images, we show the real image on the left and the paired synthetic image on the right. We find that synthetic negatives faithfully capture the "gist" and overall appearance of the object instance, but they fail to capture nuanced details such as part color and geometry, texture, and brand text/logos. These subtle differences in appearances yield an augmenting set of hard negative pairs.

C. Additional Training Details

We include additional details for all model training. In Sec. C.1, we provide an overview of GRPO [73], our main post-training method, and our training setup. We provide additional details on supervised fine-tuning (Sec. 5.2) in Sec. C.2. We include all hyperparameters in Tab. 10 and Tab. 11.

C.1. GRPO

To post-train VLMs on TWIN, we use Group Relative Policy Optimization (GRPO; [73]). We provide an overview of this method below.

Setup. Given a pair of images (I_1, I_2) with ground-truth pairwise label $y \in \{\text{yes}, \text{no}\}$, our base VLM π_θ, parametrized by θ, is prompted to produce a reasoning explanation r and final answer $\hat{y}$ whether both images depict the same instance. We wish to find π_{θ^*} that correctly identifies pairs of images that show the same instance. We include the VQA prompt used during training in Fig. 18.

Reward Design. We define a simple binary outcome reward R that compares the predicted final answer with the ground truth pairwise assignment: $R(y, \hat{y}) = \mathbf{1}_{\{y=\hat{y}\}}$. Our supervision thus relies *only* on pairwise assignments and does not use any descriptive textual annotations.

Optimization. We optimize our VLM π_θ via GRPO [73]. For each image pair (I_1, I_2) with ground truth label y, we sample G responses: $\mathcal{O} = \{(I_1, I_2, r^{(i)}, \hat{y}^{(i)})\}_{i=1}^{G}$ and compute centralized advantages:

$$A^{(i)} = R(y, \hat{y}^{(i)}) - \frac{1}{G}\sum_{j=1}^{G} R(y, \hat{y}^{(j)}) \tag{1}$$

The GRPO objective then directly maximizes expected advantages while maintaining policy stability:

$$\begin{aligned} \mathcal{L}(\theta) &= \frac{1}{G}\sum_{i=1}^{G}\Big(\hat{A}^{(i)} - \beta\,\mathrm{KL}\big[\pi_\theta \,\big\|\, \pi_{\text{ref}}\big]\Big) \\ \hat{A}^{(i)} &= \min\Big(s^{(i)}A^{(i)},\ \mathrm{clip}\big(s^{(i)},\,1-\epsilon,\,1+\epsilon\big)A^{(i)}\Big) \end{aligned} \tag{2}$$

Figure 8. **Additional Synthetic Negatives.** For each pair, we show a real image of the object instance and a synthetic image of the same instance generated with Dreambooth [70]. We find the synthetic negatives capture the "gist" of the instance, but are not faithful to fine details such as color, geometry, and texture – making them a strong additional set of hard negative pairs.

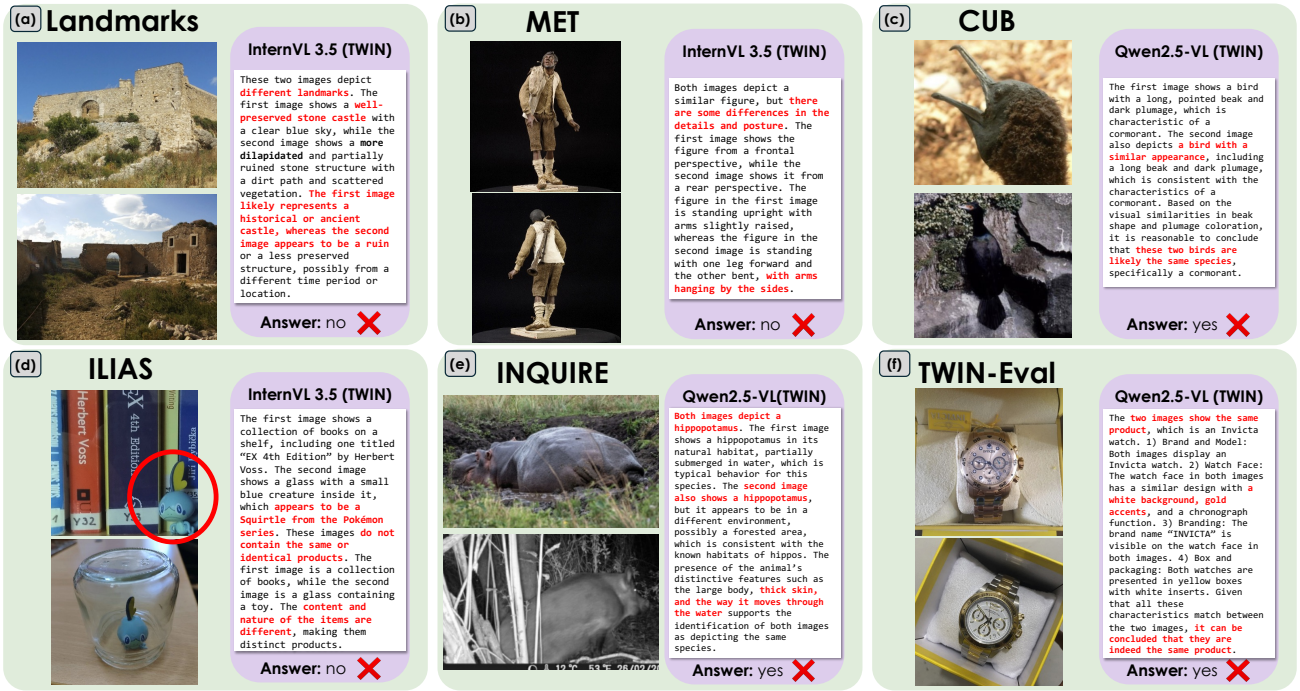

Figure 9. **Failure Cases on FGVQA for models post-trained on TWIN.** For each example, we show the source dataset, images, and model predictions. Incorrect reasoning is highlighted in **red**. The **red circle** highlights hard-to-see objects.

where $s^{(i)} = \frac{\pi_\theta(r^{(i)}, \hat{y}^{(i)} | I_1, I_2)}{\pi_{\text{old}}(r^{(i)}, \hat{y}^{(i)} | I_1, I_2)}$ represents the importance ratio, $\epsilon = 0.2$ is the clipping parameter, and $\beta = 0.01$ is the KL penalty coefficient. The formulation encourages improved fine-grained understanding while simultaneously

preventing drift from the pre-trained base policy π_{ref}.

Implementation. We train Qwen2.5-VL-3B-Instruct [5] and InternVL3.5-1B-Instruct [94] on TWIN, chosen as leading open-source VLMs at the 3B and 1B scales. We do not freeze any part of the model, and train on 4 A100 GPUs for 1 epoch. We use a batch size of 480, group size 5 and learning rate 10^{-6}. We build on the verl [76] repository for training. We detail all hyperparameters in Tab. 10. Post-training the Qwen2.5-VL 3B model for one epoch on the 560K samples of TWIN took approximately 140 hours on 4 A100 GPUs.

C.2. Supervised Fine-Tuning

We compare post-training methods in Sec. 5.2. We provide additional details for supervised fine-tuning (SFT) used in those experiments. As our model produces both an explanation and final prediction, SFT requires supervision beyond the pairwise assignment labels used in RL. To obtain high-quality supervision, we prompt Gemini-2.5-Flash [13] with samples from TWIN and retain responses where Gemini is correct. We use default generation parameters for Gemini and collect a total of 136K high-quality answers.

We train Qwen2.5-VL 3B end-to-end using SFT on the collected samples. We use LLaMa-Factory [113] for training, using a learning rate of 10^{-6} with a cosine scheduler. We train the SFT model for 2 epochs, as this yielded better accuracy on FGVQA. All hyperparameters are in Tab. 11.

D. Failure Cases

We include failure cases for models post-trained with TWIN in Fig. 9. For each example, we show the source dataset, images, and model predictions. We find that although models post-trained with TWIN demonstrate improved fine-grained understanding, they struggle with fine differences in color (*e.g.* in (a) where the ruins appears different colors due to lighting, and (f) where the watch accents differ between blue and gold). Moreover, extreme viewpoint variation, as seen in (b) remains a challenge. Lastly, we find that models struggle to extrapolate from incomplete views of the animals in (c) and (e). These challenging examples additionally highlight the difficulty of our new FGVQA benchmark suite.

InternVL on Inquire. We observe a slight decrease in performance on INQUIRE (-1.2%) when post-training InternVL3.5 1B on TWIN. We investigate the model responses and find that post-training the 1B model on TWIN reduces "direct identification" (*e.g.*, naming a species) and emphasizes comparisons of visual cues (*e.g.*, the color of the feathers). While this improves instance-level reasoning, it can fail under challenging lighting or occlusion, where the base model succeeds by relying more on priors.

E. Additional Evaluations

We provide additional evaluations for models post-trained with TWIN. In Sec. E.1 we explore the impact of post-training on TWIN on the related tasks of monument doppelganger detection and evaluate sensitivity to color and shapes. In Sec. E.2 we compare post-training to prompting with in-context-learning examples.

E.1. Additional Benchmarks

We evaluate on an additional set of benchmarks to determine if post-training on TWIN improves robustness to other image variations. We compare the baseline Qwen2.5-VL 3B model with our TWIN post-trained variant on monument doppelganger detection [8], the "yes or no" queries on shape/color sensitivity from VLMs Eye [57], and 500 pairwise queries from CUTE [34] in Tab. 8.

	Doppel. [8]	Shape Sens. [57]	Color Sens. [57]	CUTE [34]
Qwen2.5-VL 3B	50.8	56.8	96.1	58.8
+ TWIN	**55.2**	**95.7**	**99.7**	**68.0**

Table 8. **Additional benchmarks.**

Post-training on TWIN improves doppelganger detection ($+4.4\%$), despite TWIN not including monuments. This mirrors the gains on LANDMARKS and MET seen in Tab. 1. Similarly, post-trained models improve at detecting dissimilar shapes ($+38.9\%$), reach near-perfect performance at comparing colors (99.7%) and are more robust to photometric variations ($+9.2\%$ on CUTE). These results reaffirm that TWIN pushes models to attend to subtle visual cues over coarse semantic similarity.

E.2. In-Context Learning

We report results on FGVQA for in-context-learning (+ICL) prompting in Sec. E.2 to explore if we can achieve improved performance via sophisticated prompting as opposed to post-training. We randomly sample one positive and one negative pair from TWIN as in-context examples for each FGVQA query.

	Mean	TWIN-Eval	ILIAS	Landmarks	MET	CUB	Inquire
Qwen2.5-VL 3B (+ ICL)	39.5	39.0	37.3	37.5	38.5	43.5	41.2
Qwen2.5-VL 3B (+ TWIN)	**67.0**	**67.3**	**61.8**	**57.9**	**66.0**	**75.1**	**73.7**

Table 9. **In-Context Learning.**

Post-training significantly outperforms in-context-learning (67.0% vs 39.5% on average). In the ICL setting, the 3B model appears to infrequently misunderstand the order of images in the prompt, or ignores the examples entirely.

F. Comparisons to Prior Datasets

We provide a more detailed comparison to prior datasets Birds-To-Words [18] and SpotTheDiff [29]. While both of these datasets also feature pairs of images as input (similar to TWIN), there are several key differences. First, the task in these datasets is to identify the differences between

Figure 10. Examples from Birds-to-Words [18]. The dataset features 3.3K pairs of *different* bird species, along with a human-annotated description of their differences. The differences are often coarse-grained (*e.g*. Pair 3, Pair 6 – which both display significant color differences), and the text descriptions subjective (*e.g*. Pair 4 – where animal 1 is described as "more beautiful" than animal 2).

Figure 11. Examples from SpotTheDiff [29]. The dataset features 13K pairs of images taken from different frames of the same video. Unlike TWIN, that emphasizes fine differences in object texture and geometry, the differences in SpotTheDiff are primarily spatio-temporal (*e.g*. Pair 1 with the blue car leaving the frame, Pair 2 with the people moving) and highly coarse-grained (*e.g*. Pair 3).

two images, with the assumption *a priori* that the images are indeed different. In contrast, TWIN includes both negative pairs and positive pairs in which the instance is unchanged, requiring models to reason about both similarities and differences rather than only locating discrepancies. Second, the differences captured in TWIN are substantially more fine-grained, often involving subtle variations in texture or part geometry that go beyond the relatively coarse differences in prior datasets. Third, our dataset is considerably larger in scale (561K pairs vs. 13K for SpotTheDiff and 3.3K for Birds-To-Words). We provide examples and describe the differences with each individual dataset in more detail below.

F.1. Birds-To-Words

Birds-to-words [18] features 3.3K pairs of images of different bird species, along with human-annotated descriptions of the differences. We show examples from the dataset in Fig. 10. We observe that, unlike the fine-grained differences in TWIN, the differences are often much coarser (*e.g.* Pair 3 and Pair 6, which both feature significant color differences). Moreover, the human-annotated descriptions can be subjective (*e.g.* Pair 4 which describes one bird as "more beautiful").

Compared to TWIN, the dataset is orders of magnitude smaller (3.3K pairs vs the 561K in TWIN) and spans only bird species, while TWIN features a wide range of object categories with significantly different geometries (*e.g.* earbuds and mugs). Additionally, Birds-to-Words features almost exclusively *negative* pairs. While a small number of pairs are annotated as depicting the same instance, these are anomalous cases and not a core aspect of the dataset – the task is fundamentally to describe the differences between two images. In contrast, TWIN emphasizes both positive and negative pairs, requiring models to recognize not only differences but also similarities.

F.2. Comparison to SpotTheDiff

SpotTheDiff [29] comprises 13K pairs of images sourced from two different frames of the same video. The videos used depict large scenes from security camera footage. As a result, the differences between frames are predominantly spatio-temporal, concerning which objects or people are visible or no longer visible in the scene (*e.g.* a car leaving the frame, a person moving positions). We visualize examples from SpotTheDiff [29] in Fig. 11. These examples highlight that differences are generally coarse-grained and, unlike the examples in TWIN, do not require reasoning about viewpoint variation, part-level geometry, or fine-grained texture changes. Similar to Birds-to-Words, SpotTheDiff assumes the paired images are different, whereas TWIN additionally emphasizes recognizing similarities through the inclusion of positive pairs. Finally, TWIN is orders of magnitude larger in scale (561K vs. 13K pairs) and introduces challenges not present in SpotTheDiff, including substantial viewpoint variation, lighting changes, and differences in context and background.

```
Pair: Are these images of the same or identical products? For two products to be considered identical, minor changes such as those that can be explained context,
backgrounds or photography conditions are allowed, but characteristic features of the product (color, shape, size, etc.) should remain consistent. For images with
multiple products, compare only the primary product.
Explain your reasoning and then conclude with a yes or no answer in <answer> tags as <answer>yes</answer> or <answer>no</answer>.

Multi: The first image is a reference image. How many of the other images depict the same or identical products as the reference image? For two products to be
considered identical, minor changes such as those that can be explained by context, backgrounds or photography conditions are allowed, but characteristic features of
 the product (color, shape, size, etc.) should remain consistent. For images with multiple products, compare only the primary product.
Explain your reasoning and then answer with a number from 0 to 3 in <answer> tags as <answer>n</answer>.
```

Figure 12. **FGVQA Prompts:** TWIN-Eval.

```
Pair: Do these images contain the same or identical products? For two products to be considered identical, minor changes such as those that can be explained by
context, backgrounds or photography conditions are allowed, but characteristic features of the product (color, shape, size, etc.) should remain consistent.
Explain your reasoning and then answer with a yes or no answer in <answer> tags as <answer>yes</answer> or <answer>no</answer>.

Multi: The first image is a reference image. How many of the other images contain the same or identical products to one in the reference image? For two products to
be considered identical, minor changes such as those that can be explained by context, backgrounds or photography conditions are allowed, but characteristic features
 of the product (color, shape, size, etc.) should remain consistent.
Explain your reasoning and then answer with a number from 0 to 3 in <answer> tags.
```

Figure 13. **FGVQA Prompts:** ILIAS [33].

```
Pair: Do these two images contain the same landmark?
Explain your reasoning then answer with a yes or no answer in <answer> tags as <answer>yes</answer> or <answer>no</answer>.

Multi: The first image is a reference image. How many of the other images contain the same landmark as the reference image?
Explain your reasoning then answer with a number from 0 to 3 in <answer> tags.
```

Figure 14. **FGVQA Prompts:** LANDMARKS [97].

```
Pair: Do these two images contain the same piece of art?
Explain your reasoning and then answer with a yes or no answer in <answer> tags as <answer>yes</answer> or <answer>no</answer>.

Multi: The first image is a reference image. How many of the other images contain the same piece of art as the reference image?
Explain your reasoning and then answer with a number from 0 to 3 in <answer> tags.
```

Figure 15. **FGVQA Prompts:** MET [107].

```
Pair: Do these two images show a bird of the same species?
Explain your reasoning and then conclude with a yes or no answer in <answer> tags as <answer>yes</answer> or <answer>no</answer>.

Multi: The first image is a reference image. How many of the other images show a bird of the same species as the reference image?
Explain your reasoning and then conclude with a number from 0 to 3 in <answer> tags.
```

Figure 16. **FGVQA Prompts:** CUB [91].

```
Pair: Do these two images show an animal or plant of the same scientific species?
Explain your reasoning and then answer with a yes or no answer in <answer> tags as <answer>yes</answer> or <answer>no</answer>.

Multi: The first image is a reference image. How many of the other images show an animal or plant of the same scientific species as the reference image?
Explain your reasoning and then answer with a number from 0 to 3 in <answer> tags.
```

Figure 17. **FGVQA Prompts:** INQUIRE [88].

```
Are these images of the same or identical products? For two products to be considered identical, minor changes such as those that can be explained by context,
backgrounds or photography conditions are allowed, but characteristic features of the product (color, shape, size, etc.) should remain consistent. For images with
multiple products, compare only the primary product. Explain your reasoning and then conclude with a yes or no answer in <answer> tags as <answer>yes</answer> or <
answer>no</answer>.
```

Figure 18. **Prompt used for training with TWIN.**

Hyperparameter	Value
Batch size	480
Group size	5
Max prompt length	2048
Max response length	2048
Learning rate	1e-6
Optimizer	AdamW
Weight decay	0.01
KL coefficient	0.01
Clip ratio	0.2
Epochs	1
Rollout engine	vLLM
Temperature	1.0
Top-p	1.0
Gradient clipping	Max norm 1.0
Freeze vision tower	False
Mixed precision	bf16

Table 10. Hyperparameters for GRPO training.

Hyperparameter	Value
Batch size	256
Max prompt length	2048
Max response length	2048
Learning rate	1e-6
Optimizer	AdamW
Scheduler	Cosine
Warmup ratio	0.1
Epochs	2
Deepspeed config	ZeRO Stage 3
Gradient clipping	auto
Freeze vision tower	False
Mixed precision	bf16

Table 11. Hyperparameters for SFT training.