Tiny Inference-Time Scaling with Latent Verifiers

Supplementary Material

A. Additional Implementation Details

A.1. Dataset Construction

We build datasets pairing DiT embeddings with task-specific targets: captions for the alignment phase and binary labels for fine-tuning. In the following, we detail this generation process for both training phases of VHS.

Alignment Dataset. Due to the noisiness of the image-text couples of the LLaVA pre-training dataset [26], we use Gemma3-4B [42] to produce refined captions as prompts. We then generate images, extract the corresponding DiT embeddings, and re-caption the outputs to guarantee alignment.

Verifier Fine-Tuning Dataset. We leverage the prompt set from ReflectDiT [21] to synthesize a new image corpus, generating 20 variants per prompt and extracting their internal activations. To establish ground-truth labels for these samples, we implement an automated annotation pipeline using Gemma3-4B. Specifically, the LLM generates the metadata required for the GenEval [10] verification pipeline by first classifying the prompt tag (*e.g.*, single object, two objects, counting, position, colors, attribute binding) and subsequently deriving the specific inclusion and exclusion lists. Finally, the GenEval verifier processes the synthesized images with this metadata to assign a binary label to each sample. The LLM prompts for data generation are shown in Fig. 10.

A.2. Architectural Details

Hidden Features Extraction. We extract hidden layer features from the output of the $\ell^\star$ layer and normalize them using mean and variance statistics pre-computed over a 50k-example subset of the alignment training set. For SANA-Sprint at standard resolution, this process yields a representation of 1024 spatial features, each with a hidden dimensionality of 2240. Similarly, PixArt-α-DMD produces a feature map of the same spatial size (1024 features), but with a smaller hidden dimensionality of 1152.

AE Features Extraction. AE features are extracted by flattening the output latents of the generator across spatial dimensions. For SANA-Sprint, the deep compression autoencoder reduces spatial resolution by 32×. For a standard 1024×1024 input resolution, this produces a 32×32 feature grid with a dimensionality of 32, which, after flattening and projection, results in 1024 image tokens as input to the LLM. Conversely, for PixArt-α-DMD, the generator uses a standard autoencoder with KL loss [34]. Given a 512×512 input resolution, this process yields a 64×64 feature grid, which we flatten and project into the LLM embedding space, producing 4096 image tokens. This high token count increases the inference latency of the LLM further rendering AE features unfeasible for our task.

Multimodal Connector. The connector module is implemented as a two-layer MLP with a GELU activation function [14], with the first projection layer bringing the features to LLM-compatible dimensionality.

LLM Setup. The full prompt given to the verifier for evaluating images can be seen in Fig. 10, allowing the evaluation procedure to be reproduced.

A.3. Training Details

Alignment Stage. The model is trained for one epoch on the 558k-sample alignment dataset. We use a total batch size of 512 distributed across 16 NVIDIA A100 GPUs. The learning rate follows a cosine schedule, peaking at 1×10^{-3}, following a warm-up phase lasting 3% of the total training steps, with the AdamW [28] optimizer.

Verifier Fine-Tuning. On the other hand, the fine-tuning stage of our pipeline is carried out using a total batch-size of 64 on 8 NVIDIA A100 GPUs, a cosine learning rate scheduling with a maximum value of 2×10^{-5}, and the AdamW optimizer. We select the model yielding the best evaluation loss during a 10 epochs training. To address class imbalance in the SANA-Sprint training set, we employ a weighted cross-entropy loss, assigning weights of 0.37 to the positive class and 0.63 to the negative class, following sample distribution. Conversely, for the PixArt-α-DMD experiments, equal weights are utilized to reflect the balanced class distribution of the dataset. Moreover, the focal loss ablation study follows the formulation proposed in [23]:

$$\mathrm{FL}(p_\mathrm{t}) = -\alpha_\mathrm{t}(1-p_\mathrm{t})^\gamma \log(p_\mathrm{t}), \tag{4}$$

where p_t represents the estimated probability of the model for the ground-truth class, α_t is a weighting factor assigned to each class to counteract class imbalance, and γ modulates the loss function by exponentially down-weighting "easy" examples, forcing the model to concentrate on "hard", misclassified examples that contribute the most to the training error. We set α as the class imbalance factor and γ to 2.

A.4. Inference Setup

Score Computation. Following [47], we convert binary verifier outputs into a continuous score. The score is defined as the predicted probability of the verifier for the sampled token, negated if it belongs to the negative class (*i.e.*, "no"). Otherwise (*i.e.*, "yes"), the probability is used as is.

Best-Of-N Setup. Fig. 4 shows the Best-of-N setup: each candidate is passed through the first DiT layers and scored

Table 5. Accuracy (%) of SANA-Sprint [7] on the GenEval benchmark [10]. Ablation on score methodology on a time budget of 1100 ms.

Verifier	Single	Two	Counting	Color	Position	Attribution	Overall
VHS							
w/ h_7 No Scoring	99.3	90.7	57.5	87.5	63.2	52.6	74.7
w/ h_7 Token Probability Scoring **(Ours)**	**100.0**	**95.7**	**66.5**	88.9	**69.8**	**63.8**	**80.5**
MLLM w/ CLIP							
w/ No Scoring	99.5	91.1	58.3	88.3	62.0	54.8	75.3
w/ Token Probability Scoring	**100.0**	92.7	66.0	88.9	65.9	61.6	78.8
MLLM w/ AE							
w/ No Scoring	99.7	87.5	56.5	88.7	55.0	49.2	72.2
w/ Token Probability Scoring	99.7	90.7	61.3	**90.8**	59.6	49.3	74.7

Table 6. GenEval benchmark [10] results with compute-equivalent baselines.

Budget	Generator	Steps	Verifier	BoN	Single	Two	Count	Color	Pos	Attr	All
550ms	SD3.5-Turbo	1	MLLM w/ CLIP	Bo1	83.0	29.9	35.5	61.1	11.4	13.6	37.4
	Flux-Schnell	1	MLLM w/ CLIP	Bo1	99.0	90.3	60.5	82.8	28.0	**58.4**	68.9
	Sana-Sprint	1	**VHS (Ours)**	Bo4	**100.0**	**93.9**	**61.5**	**90.6**	**66.2**	**58.4**	**78.1**
1100ms	SD3.5-Turbo	1	MLLM w/ CLIP	Bo2	90.1	35.6	37.8	63.8	13.8	16.6	41.2
	Flux-Schnell	1	MLLM w/ CLIP	Bo2	99.8	94.3	65.8	84.7	34.4	**65.0**	73.2
	Flux-Schnell	2	MLLM w/ CLIP	Bo2	99.0	91.7	61.3	80.9	28.4	57.4	68.9
	SANA-Sprint	1	**VHS (Ours)**	Bo9	**100.0**	**95.7**	**66.5**	**88.9**	**69.8**	63.8	**80.5**
1650ms	SD3.5-Turbo	1	MLLM w/ CLIP	Bo3	91.3	42.6	40.3	64.8	16.6	19.4	44.1
	Flux-Schnell	1	MLLM w/ CLIP	Bo3	**100.0**	95.2	65.8	84.7	39.8	**66.0**	74.7
	Flux-Schnell	2	MLLM w/ CLIP	Bo3	**100.0**	94.3	**68.0**	84.3	36.0	62.4	73.3
	SANA-Sprint	1	**VHS (Ours)**	Bo15	**100.0**	**96.0**	67.3	**89.1**	**70.4**	64.6	**80.9**

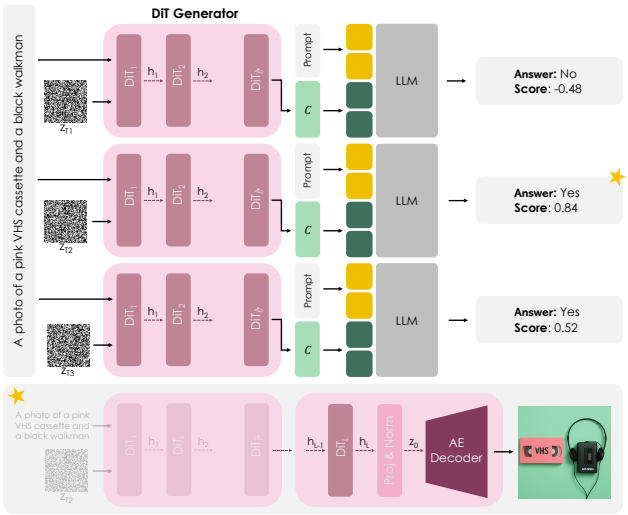

Figure 4. Efficient Best-of-N pipeline with VHS.

by the LLM, after which the highest-scoring candidate completes the remaining layers and is decoded to pixel space.

Inference and Evaluation Parameters. To run inference on the SANA-Sprint model, we stick to the standard resolution of 1024×1024. We employ a CFG of 1.0, and run the generator in a single step. We average our results over 5 different seeds to get more stable estimations. Moreover, for PixArt-α-DMD, the resolution is set 512×512.

B. Additional Analyses and Experiments

B.1. Data Quality Analysis

To verify data quality for the alignment stage of VHS training, we evaluate the multimodal consistency of our generated image–caption pairs using CLIP-Score [15]. Our pipeline, which re-captions the original images, synthesizing new images, and re-captioning the outputs, produces pairs with a CLIP-Score of 73.5. Compared to the original dataset's score of 70.5, this demonstrates that our approach not only preserves semantic alignment, but actually improves image–text consistency over the original LLaVA alignment data.

B.2. Additional Studies on the Continuous Score

Ablation on the Continuous Score. In Table 5, we present an ablation study to validate the use of token probabilities as continuous scores for image selection. Specifically, we compare GenEval accuracies under an 1100 ms budget with SANA-Sprint using two distinct strategies: (i) random selection among images classified as positive ("yes") by the verifier, and (ii) selection of the highest-scoring image based on token probability, as described in Sec. A.4. We observe that utilizing probability scores consistently enhances accuracy across all categories and verifier types, yielding an overall improvement of up to 5.8% for VHS. This suggests that token probabilities provide a granular measure of picture quality that binary decisions alone fail to capture.

Table 7. Accuracy (%) of SANA-Sprint [7] on the GenEval benchmark [10], varying levels of LoRA Fine-Tuning on a time budget of 1100 ms for MLLM w/ CLIP and VHS.

Verifier	Single	Two	Counting	Color	Position	Attribution	Overall	Δ
No LoRA Fine-Tuning								
MLLM w/ CLIP	**100.0**	94.7	59.3	**91.9**	69.4	60.2	79.1	+1.4
VHS	**100.0**	**95.7**	**66.5**	88.9	**69.8**	**63.8**	**80.5**	
50% LoRA Fine-Tuning								
MLLM w/ CLIP	**100.0**	**92.9**	60.8	**90.6**	63.4	58.2	77.3	+1.8
VHS	**100.0**	92.1	**61.5**	89.8	**70.4**	**62.0**	**79.1**	
100% LoRA Fine-Tuning								
MLLM w/ CLIP	99.5	88.3	57.3	87.9	57.2	**56.2**	73.9	+2.4
VHS	**100.0**	**91.1**	**59.5**	**89.4**	**64.4**	56.0	**76.3**	

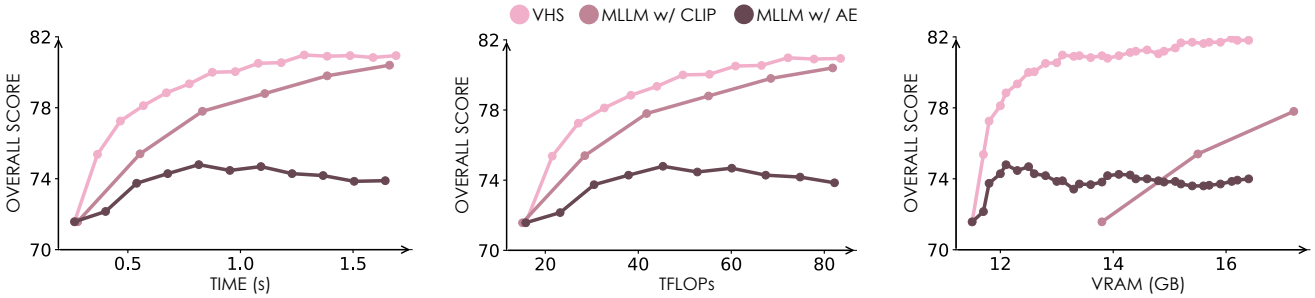

Figure 5. Overall accuracy (%) of SANA-Sprint [7] on GenEval [10] across time (seconds) TFLOPs, and VRAM usage (GB).

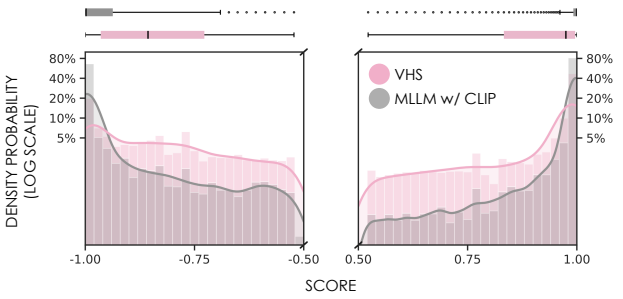

Figure 6. Score distribution comparison between MLLM w/ CLIP and VHS. Histograms display the frequency of scores within 5% bins, with smoothed density curves overlaid in gray (MLLM w/ CLIP) and pink (VHS). Box plots above show the quartile ranges and distributional characteristics of each verifier.

Score Distribution Analysis. In Fig. 6, we analyze the score distributions produced by VHS and MLLM w/ CLIP on the prompts of the GenEval benchmark, using 32 generations per prompt with SANA-Sprint. The distributions reveal that VHS produces less extreme scores compared to MLLM w/ CLIP. Specifically, MLLM w/ CLIP assigns roughly 80% of samples to the extreme score ranges ([0.95, 1.0] or [-1.0, -0.95]), indicating a strong tendency toward binary judgments. In contrast, VHS assigns only about 40% and 20% of samples to the positive and negative extremes, while distributing a larger proportion of scores toward the center of the range. This more balanced distribution reflects the ability of VHS to assign more spread-out scores, enabling finer-grained discrimination between samples of varying quality.

B.3. Comparison with additional baselines

In order to assess the trade-offs and resource allocation strategies within the Best-of-N pipeline under tight time budgets, in Table 6 we present additional comparisons between VHS and various baselines equipped with different generators, although not directly comparable. Specifically, we compare against FLUX.1-Schnell [3] and Stable Diffusion 3.5-Turbo[40], exploring different Best-of-N and multi-step configurations under equivalent budgets of 650 ms, 1100 ms, and 1650 ms. Overall, VHS achieves the best results across all evaluated time budgets, as configurations relying on larger samplers are forced to operate with too few denoising steps or single-shot selection, limiting their effectiveness. Finally, Fig. 5 presents an extended comparison of VHS and the two baselines, plotting overall GenEval score as a function of three computational budget metrics: inference time, TFLOPs, and VRAM usage (GB), demonstrating clear advantages for VHS across all three.

B.4. Robustness to Model Updates

Acknowledging the tight coupling VHS introduces between the generator and its latent verifier, we investigate how distribution shifts in the generator's weights affect verification quality. To assess this, we fine-tune SANA-Sprint on an aesthetic dataset [45] using LoRA [17] and measure GenEval performance within an 1100 ms budget. Although distribution shift degrades overall performance, VHS exhibits significantly greater resilience than the pixel-space verifier (MLLM w/ CLIP), suggesting that the generator's hidden

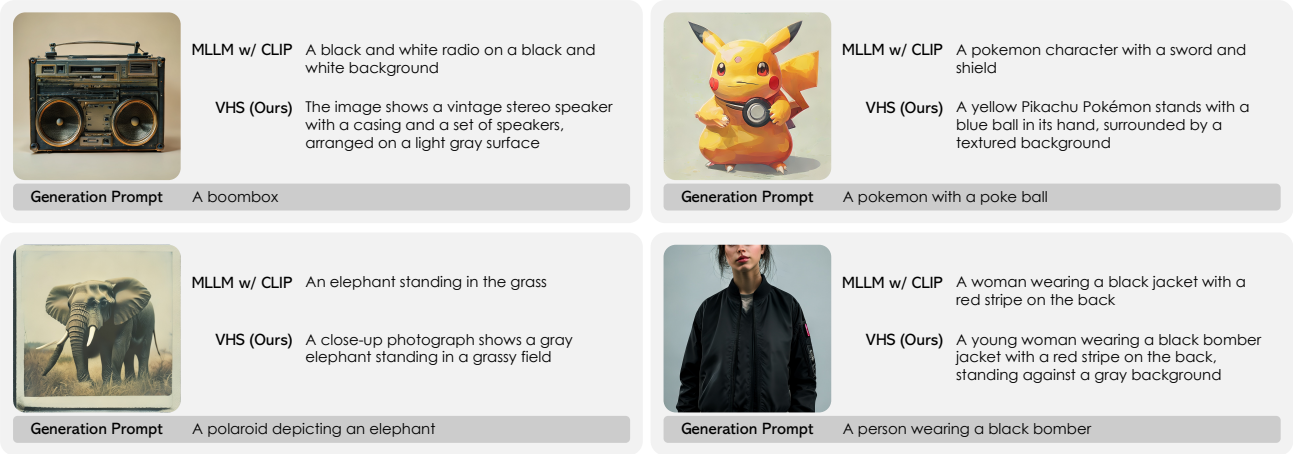

Figure 7. Qualitative results of captioning produced by VHS and MLLM w/ CLIP after the alignment training phase.

Figure 8. Cases where judgments from VHS is not aligned with the GenEval verifier.

states remain semantically stable during fine-tuning. This relative stability is reflected throughout the training process. As observed in Table 7, the performance gap between the methods grows from 1.4% at the baseline to 1.8% at mid-epoch, reaching 2.4% after a full epoch.

C. Additional Qualitative Results

In Fig. 9 we provide a qualitative evaluation of VHS using sample prompts from the GenEval benchmark with the SANA-Sprint generator, emphasizing its effectiveness in various challenging scenarios. VHS captures fine-grained details, distinguishing accurate images from those that are subtly incorrect. For instance, in the counting task involving "four giraffes", VHS correctly identifies the discrepancy in the image, which contains five giraffes, assigning a negative score consistent with the GenEval verifier. Similarly, VHS shows good performance in spatial reasoning, as it correctly validates the "baseball glove right of a bear" example which the baseline falsely rejects. These results further confirm that VHS accurately validates correct generations related to counting, attribute binding, and spatial relationships.

Moreover, Fig. 7 presents qualitative examples of captions produced by VHS following only the alignment training phase. For each example, we generate sample images using the specified generation prompts and extract the corresponding hidden layer activations to feed into VHS. The examples show that VHS consistently produces accurate and detailed captions across diverse subjects. For instance, given the generation prompt "A pokemon with a poke ball", the generator produces an image which VHS then captions with a comprehensive description, identifying the character as "a yellow Pikachu Pokémon", and noting the "textured background". Similarly, when the generator creates an image from the prompt "A boombox", VHS correctly describes it as "a vintage stereo speaker with a casing and a set of speakers, arranged on a light gray surface", capturing fine-grained visual details such as the surface color and arrangement.

D. Limitations

While our proposed method demonstrates robust performance, we acknowledge certain limitations. Most notably, the verifier is coupled to the underlying generator architecture, as it operates directly on its hidden representations. This tight integration enables substantial efficiency gains and strong semantic alignment, but limits out-of-the-box transferability across substantially different generative backbones. In our experiments, we observe stable behavior under incremental model updates (Table 7); however, if the architecture changes significantly, VHS must be retrained to maintain compatibility and verification performance. In practice, this constraint is of limited relevance in typical production settings, where generators are updated incrementally.

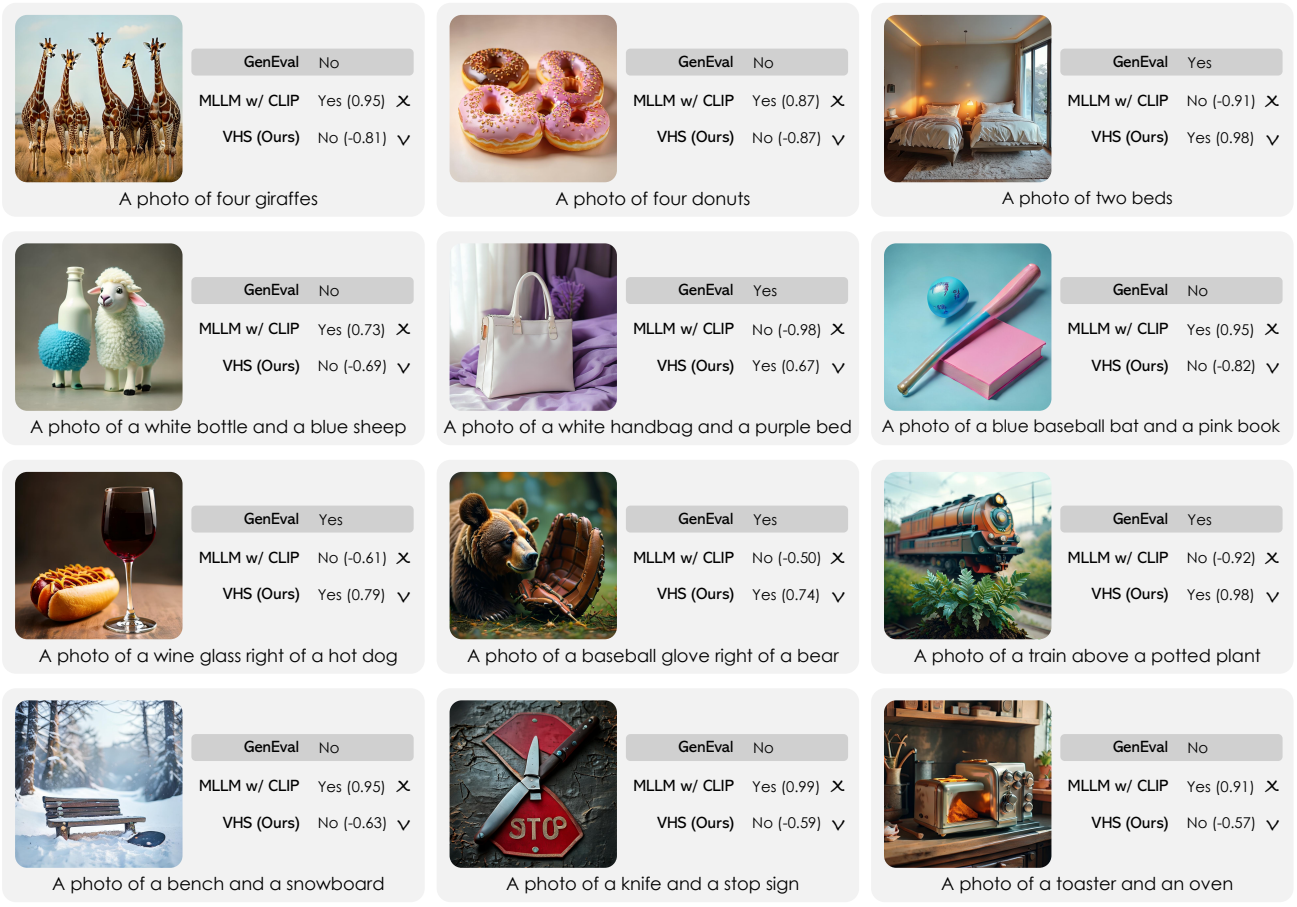

Figure 9. Visual comparison of the best pick images by different verifiers for images generated by SANA-Sprint [7] on GenEval [10] prompts. ✓ and × express the alignment with the GenEval verifier.

1. Prompt for Tag Generation

```
You are an assistant that classifies image generation prompts into one of six categories.
Given a text prompt describing an image, output ONLY the corresponding tag from this list:
- single_object
- two_object
- counting
- colors
- position
- color_attr

Rules:
1. Respond with exactly one of the tags above — nothing else.
2. Classification criteria:
   - "single_object" → only one object is mentioned.
   - "two_object" → exactly two different objects are mentioned (e.g. "a cat and a dog").
   - "counting" → a specific number of identical objects is requested (e.g. "three dogs").
   - "colors" → a single object with a color attribute (e.g. "a purple umbrella").
   - "position" → objects are described with a spatial relation (e.g. "a cat below a table", "a
   man left of a horse").
   - "color_attr" → two or more different objects, each with their own color (e.g. "a red apple
   and a green pear").
3. If uncertain, choose the most specific applicable tag.

Examples:
Input: "a photo of an umbrella" → single_object
Input: "a photo of a bowl and a pizza" → two_object
Input: "a photo of three persons" → counting
Input: "a photo of a purple hair drier" → colors
Input: "a photo of a couch below a cup" → position
Input: "a photo of a red skis and a brown tie" → color_attr

The prompt is {input_prompt}.
```

2. Prompt for Captioning on the Alignment Set

```
Describe the image in one concise sentence. Be objective and precise, without speculation.
Output only the description in plain text, without line breaks.
```

3. Prompt for Image Scoring

```
You are an AI assistant specializing in image analysis and ranking. Your task is to analyze and
compare image based on how well they match the given prompt.
<image> The given prompt is: {input_prompt}.
Please consider the prompt and the image to make a decision and response directly with 'yes' or
'no'.
```

Figure 10. Prompts employed for dataset generation and image scoring.