Appendix

A. Related Work

Text-to-Image Diffusion Models. Text-to-image diffusion models have rapidly advanced in terms of model architecture and training strategies. In terms of architecture evolution, the community has transitioned from the prevalent U-Net diffusion models [45] towards diffusion transformers (DiT) [38]. Regarding training strategies, earlier complex hand-designed diffusion schedules [19, 51] have given way to simpler flow-based formulations [31, 34], significantly enhancing training efficiency through techniques such as multi-resolution progressive training [9]. Recently, scaling up both the training datasets and model parameters has led to the emergence of various large-scale, flow-based diffusion transformer models [12, 14, 27, 42, 60, 68].

Diffusion Inference-Time Enhancement. Building upon the powerful pretrained diffusion models, recent research has increasingly focused on unleashing their potential at inference time. One line of research has observed that the initial noise significantly impacts generation quality [41, 67], prompting methods to identify superior initialization strategies [2, 35, 41, 67]. Another research direction aims to improve the iterative sampling procedure of diffusion models [3, 48, 65], notably through denoising and inversion [50]. Additionally, recent studies [29, 60] demonstrate that augmenting input prompts can substantially improve visual fidelity and text-image alignment. While existing methods focus on parallel single-pass generation, our work proposes a sequential generation then refinement procedure, integrating both parallel and sequential inference-time scaling into a unified framework.

Scaling Inference-Time Compute. Recent studies on LLMs have provided valuable insights into inference-time scaling laws. A primary line of investigation [13, 63] has explored search algorithms, such as best-of-N and beam search, with verifiers to select higher-quality outputs. Another prominent direction focuses on enabling LLMs to refine their own outputs. For instance, techniques [24, 36, 47] such as zero-shot prompting have been employed to elicit self-reflection from models, enabling them to iteratively enhance their outputs. Furthermore, supervised fine-tuning (SFT) and reinforcement learning (RL) approaches [11, 17, 26, 43, 57] have also been introduced to explicitly train models for reflective self-improvement. One recent work [16, 22] investigated RL, test-time scaling, and reflection on autoregressive image generation models, providing preliminary insights into this field. Meta also presented a comprehensive framework [49] that systematically unifies these directions, which thoroughly investigates the trade-offs between the pretraining scale and the computation of inference time, significantly inspiring our approach.

B. GenRef Dataset

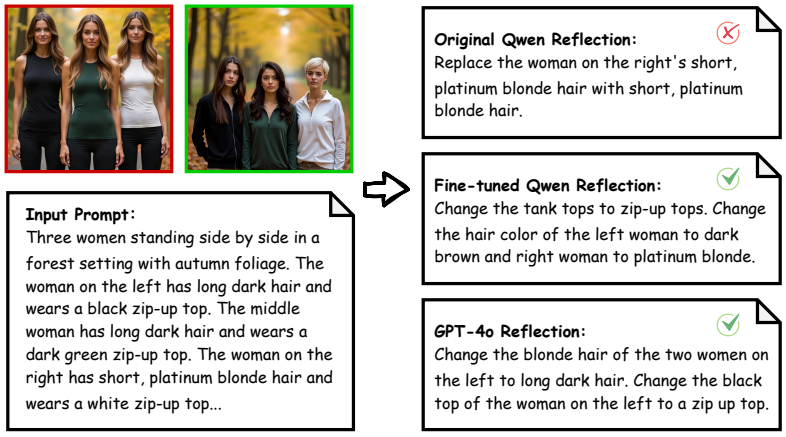


Figure 6. Comparisons of textual reflection generated by original Qwen2.5-VL-7B, our fine-tuned image reflector, and GPT-4o.

Reflection Annotation. After constructing diverse image pairs, it is essential to annotate them with textual reflections that explicitly describe how to transform a flawed image into its corresponding higher-quality counterpart. We experiment with various MLLMs and observed that even the current state-of-the-art open-source model, Qwen2.5-VL [4], tends to generate inaccurate reflections when prompted in a zero-shot manner, exhibiting severe hallucinations as illustrated in Fig. 6. Therefore, we leverage closed-source model APIs [21, 53] and design a CoT image reflection annotation pipeline, enabling

the models to step-by-step analyze image pairs. Specifically, we concatenate two images into an image grid and provide it as input together with the prompt. The model first identifies key differences between the two images, then makes a judgment regarding image preference, and finally produces a concise reflection instructing how to correct the identified flaws in the lower-quality image. The detailed CoT prompts are provided in Fig. 21 in the Appendix.

Through this CoT-based annotation approach, we observe a significant improvement in the accuracy and reliability of generated reflections. Moreover, intermediate results from the reasoning process, such as the explicit image preferences, can also serve as valuable annotations for reward model training, as discussed in the next paragraph. We annotate approximately 270K CoT reflections using this annotation pipeline, and after careful filtering and quality control, we obtain a final dataset of 227K high-quality CoT reflections, named GenRef-CoT. Subsequently, we fine-tune the Qwen2.5-VL-7B model on this curated reflection dataset, enabling it to annotate the full dataset comprehensively[3]. Fig. 6 illustrates a qualitative comparison among reflections generated by the original Qwen2.5-VL-7B, our fine-tuned reflector, and GPT-4o, clearly demonstrating that our fine-tuned model produces substantially improved and more accurate reflections compared to the original model. We also visualize the word cloud of reflections in Fig. 2, which shows the pattern of executable instructions.

Verifier Post-processing. To further ensure the quality of our dataset, particularly regarding the quality gap between paired images, we train an image reward model (verifier) to quantitatively evaluate image quality. Specifically, we leverage the intermediate image preference annotations from GenRef-CoT and selected pairs whose preferences align consistently with the image pair annotations in GenRef, serving as confident, high-quality preference data. Inspired by recent advancements in reward modeling for video generation [33], we adopt the Bradley–Terry (BT) pairwise comparison approach [6] to train our image reward model. The BT framework utilizes a pairwise log-likelihood loss to explicitly model the reward gap between preferred and non-preferred image pairs, defined as:

$$\mathcal{L}_{\mathrm{BT}} = -\mathbb{E}_{(y,x_w,x_l)\sim D}\left[\log \sigma\left(r_\eta(x_w, y) - r_\eta(x_l, y)\right)\right], \tag{4}$$

where y denotes the input prompt, (x_w, x_l) represents the preferred and non-preferred image pair respectively, r_η is the learnable reward model, and $\sigma(\cdot)$ refers to the logistic sigmoid function. Leveraging this verifier, we conduct a rigorous post-processing step on our dataset, applying multiple criteria to filter out lower-quality data samples. Furthermore, the trained verifier can also be utilized for inference-time scaling, which we elaborate in detail in Section 3.2.

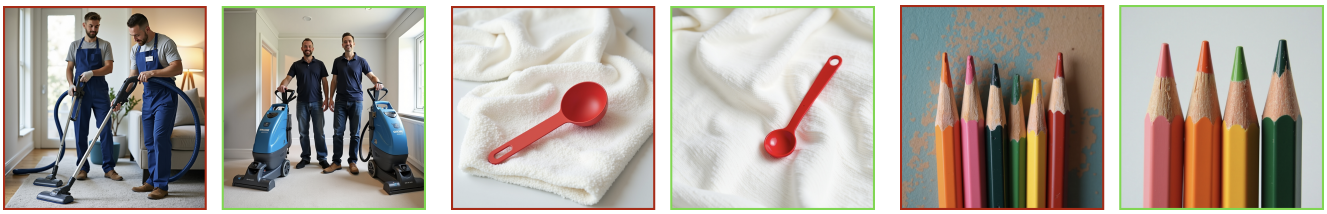
Dataset Preview. In Fig. 7, Fig. 8, and Fig. 9, we provide samples from our GenRef dataset. We divide these figures with respect to the subsets we used during data curation, except for the "edit" samples that were sourced from the OmniEdit dataset [56]. For each image, we provide the prompt and reflection pairs. The red and green borders indicate the starting and final images, respectively. For best viewing experience, we recommend zooming in.

(a) (b) (c)

Figure 7. Samples from the pool of reward-based data. (**a**) **Prompt**: *A surreal digital illustration of a floating island with a large, lush tree and cascading waterfalls, set against a twilight sky with a crescent moon, surrounded by misty mountains and vibrant, ethereal colors*. **Reflection**: *Remove the sun in the background. Add more mist around the mountains*. (**b**) **Prompt**: *A surreal digital artwork depicting a clear glass resting on rocks, containing a miniature landscape with a river, pine trees, and a sunset, surrounded by pink flowers, creating a dreamlike, photorealistic scene with vibrant colors and intricate details*. **Reflection**: *Add more rocks around the glass. Add more pink flowers around the glass*. (**c**) **Prompt**: *A young woman with long red hair ice skates gracefully on a frozen lake, surrounded by snow-covered mountains and evergreen trees, creating a serene and ethereal winter scene*. **Reflection**: *Change the woman's outfit to a white sweater and skirt. Make the woman skate more gracefully*.

Fig. 10 shows samples from the dataset created for fine-tuning our Qwen model, as described in Section 2.2. The green and red borders denote the correct and incorrect images (as deemed by closed-source APIs), respectively.

[3] Appendix B shows a few samples from this dataset.



(a) (b) (c)

Figure 8. Samples from the pool of rule-based data. (**a**) **Prompt**: *a photo of two carpet cleaners*. **Reflection**: *Replace the vacuum cleaners with professional-grade carpet cleaning machines and adjust the posture of the individuals to face forward while holding the machines*. (**b**) **Prompt**: *a photo of a white blanket and a red measuring spoon*. **Reflection**: *Change the texture of the blanket to a smooth, woven pattern instead of a fluffy one*. (**c**) **Prompt**: *a photo of four colored pencils*. **Reflection**: *Remove two pencils from the group to leave only four pencils visible*.

(a) (b) (c)

Figure 9. Samples from the pool of long-short prompt data. (**a**) **Prompt**: *Portrait of a middle-aged man with a beard, seated indoors, looking slightly to the right. He wears a dark blue shirt and is positioned in the lower left of the frame. His right hand holds a newspaper, partially visible in the foreground. The background features rustic wooden walls with a warm, weathered texture, and a wooden mirror frame is partially visible on the right. The lighting is soft and diffused, casting gentle shadows on his face, creating a contemplative mood. The color palette is muted with earthy tones, emphasizing a cozy, intimate atmosphere. The composition is balanced, with a shallow depth of field that keeps the focus on the man's expression. Photorealistic, cinematic, warm, introspective, visually balanced*. **Reflection**: *Change the suit jacket into a dark blue shirt. Remove the gray sweater vest*. (**b**) **Prompt**: *Studio portrait of a young woman with fair skin and long, wavy red hair, centered against a dark grey background. She gazes directly at the camera with a neutral expression, her lips painted a vibrant red. Her right hand is raised to her chin, with fingers gently touching her cheek. She wears a crisp white blouse with a statement necklace featuring large, dark blue gemstones. The lighting is soft and even, highlighting her freckles and the texture of her hair. High contrast, sharp focus, professional studio photography, neutral color palette, elegant and poised, classic portrait composition*. **Reflection**: *Paint the lips a vibrant red. Replace the necklace with one that features large, dark blue gemstones*. (**c**) **Prompt**: *A striking portrait of an elderly woman dressed as a superhero in an urban setting. She stands confidently in the foreground, wearing a red helmet with a visor, green eye mask, and a red and blue superhero costume with a cape. Her right hand is raised, adorned with a silver glove, while her left hand rests on her hip, also gloved. A can of food is strapped to her left arm. The background features a city street with a yellow traffic light and a bus with an American flag on its side, parked on the right. A brick building with the sign "BRINKLEY'S" is visible behind her. The scene is set in a bustling city environment with blurred buildings and a taxi in the distance. Photorealistic, high contrast, dramatic lighting, vibrant color palette, sharp focus on the subject, urban superhero theme, dynamic composition, slightly desaturated background, cinematic feel*. **Reflection**: *Change the woman's outfit to a red and blue superhero costume with a cape. Add a red helmet with a visor, green eye mask, and a silver glove to the woman's hand*.

(a) (b) (c)

Figure 10. Samples from the pool of CoT (chain-of-thought) data. (**a**) **Prompt**: *Dimly lit photograph of an empty booth in a cozy, vintage-style diner. Red leather benches form an L-shape around a pair of adjoining hexagonal marble tables in the left center against the back wall. The tables, with dark veining and brown wooden edges, sit atop black metal bases. The wall shows a textured, weathered green paint over a brick surface, subtly shadowed. In the foreground, a waist-high partition with vertical ribbed glass edges frames the scene on the right. Flooring includes worn terrazzo tiles with visible dust. Warm, ambient lighting creates an inviting yet mysterious atmosphere. Photorealistic, shallow depth of field, warm tones, nostalgic, intimate mood, reminiscent of mid-20th-century design*. **Analysis**: *The left image better matches the prompt. It features a dimly lit, cozy atmosphere with red leather benches forming an L-shape around two adjoining hexagonal marble tables. The wall has a textured green paint over a brick surface, and the terrazzo flooring appears worn and dusty. The right image deviates from the prompt as the benches form a U-shape instead of an L-shape, the wall includes wooden paneling not mentioned in the prompt, and the lighting is less dim and warm, reducing the nostalgic and intimate mood*. **Instruction**: *Change the U-shaped bench arrangement to an L-shape. Remove the wooden paneling on the walls and replace it with textured green paint over a brick surface*. (**b**) **Prompt**: *A minimalist, flat-design illustration of stylized Arabic calligraphy forming a coffee pot with steam, set against a warm brown background, featuring a cream color palette and subtle shadows for depth*. **Analysis**: *The left image accurately portrays stylized Arabic calligraphy forming a coffee pot and steam, while the right image only contains Arabic calligraphy around a regular coffee pot. The prompt specifically calls for the Arabic calligraphy to form the pot and steam*. **Instruction**: *Transform the Arabic calligraphy into the form of a coffee pot with steam. Remove the existing regular coffee pot*. (**c**) **Prompt**: *A man in a red and yellow plaid shirt sits on a floral sofa in a warmly lit, cozy living room with natural light, a bookshelf, and a floral lamp, creating a relaxed and intimate atmosphere*. **Analysis**: *The right image is better because it depicts a man in a red and yellow plaid shirt seated on a floral sofa within a warmly lit, cozy living room, effectively conveying a relaxed and intimate setting as described in the prompt. The left image features a man in a plaid shirt, but the colors are not quite right, and the setting is not as warm or cozy, and the lighting is not right*. **Instruction**: *Change the shirt color to red and yellow plaid. Add natural light to give the setting a warm and cozy feel*.

C. ReflectionFlow Algorithm

The proposed ReflectionFlow framework is as follows:

Algorithm 1 The proposed ReflectionFlow framework

Require: prompt y, generator G_θ, corrector C_ϕ, verifier V, scaling width N, scaling depth M
Ensure: High-quality image that best realizes user intent

$X_0 \leftarrow \emptyset$ ▷ Initial image set
for $j = 1$ to N **do**
  Sample $z^j \sim \mathcal{N}(0, I)$
  $x_0^j \leftarrow G_\theta(y, z^j)$ ▷ Generate initial image
  $X_0 \leftarrow X_0 \cup \{x_0^j\}$
end for
for $i = 1$ to M **do**
  $s_i \leftarrow V(X_{i-1}, y)$ ▷ Score previous images
  $X_i \leftarrow \emptyset$
  for $j = 1$ to N **do**
    $r_i^j, y_i^j \leftarrow V(x_{i-1}^j, y, s_i)$ ▷ Generate reflection
    $x_i^j \leftarrow C_\phi(y_i^j, r_i^j, x_{i-1}^j)$ ▷ Refine with corrector
    $X_i \leftarrow X_i \cup \{x_i^j\}$
  end for
end for
return $\arg\max_{x \in \bigcup_{i=0}^{M} \bigcup_{j=1}^{N} x_i^j} V(x, y)$

D. Additional Experiments

Efficiency-Aware Comparisons. We present additional ablations in Tab. 2 and Fig. 4, comparing our method with SFT-based approaches and naive scaling of diffusion steps under fair conditions. ReflectionFlow consistently outperforms these baselines, demonstrating the effectiveness of our triplet-based reflection tuning in enabling iterative reasoning. As analyzed in Section 2.1, decomposing tasks into a generate-and-refine paradigm substantially reduces learning difficulty, allowing the model to generalize efficiently with high-quality training data, which cannot be attained by conventional SFT methods. This is also validated in recent works [35] and motivated us to rethink the dimensions of inference-time scaling for diffusion models in our paper. Our GenRef dataset is compatible with methods like DPO and SFT, and we report standard SFT results with GenRef in Tab. 2. Note that our reflection tuning only requires about 10 hours of training on an 8×A100 node, yielding negligible amortized inference overhead.

Our correction model is initialized from the base generation model and therefore has an identical parameter count. Tab. 2 reports the wall-clock time for ReflectionFlow and baselines (measured as the time of a single inference-time scaling iteration). Our framework uses 1.5× the runtime of baselines while achieving substantially better results. Currently, we use OpenAI's API for prompt refinement, which makes direct FLOPs calculation infeasible and contributes most of the extra time cost. This overhead could be significantly reduced by switching to open-source MLLMs [4].

Method	N=1	N=2	N=4	N=8	N=16	Latency
FLUX.1-dev	0.67	0.74	0.78	0.82	0.83	**36.03 sec/iter**
+ SFT	0.66	0.72	0.76	0.80	0.83	36.03 sec/iter
+ Reflection SFT	0.69	0.72	0.78	0.82	0.84	36.03 sec/iter
+ Prompt Scaling	0.69	0.72	0.80	0.84	0.85	44.28 sec/iter
+ ReflectionFlow	**0.77**	**0.85**	**0.87**	**0.88**	**0.90**	55.45 sec/iter

Table 2. Performance and efficiency comparisons.

Iterative Refinement Strategies. We systematically investigate different exploration strategies within the ReflectionFlow framework. Specifically, we conduct ablation experiments by varying two key hyperparameters: search width N, which denotes the number of candidate images generated at each iteration, and reflection depth M, representing the number of iterative

refinement rounds. Given a fixed computational budget of $N \times M = 16$, we explore three different refinement strategies: (1) *Sequential*, generating one candidate per refinement step; (2) *Parallel*, generating multiple candidates simultaneously per iteration; and (3) *Combine*, balancing both depth and breadth by moderately expanding candidate branches per iteration. We use GPT-4o as the verifier.

Tab. 3 clearly shows that the sequential strategy (N1M16) achieves the highest overall performance of 0.78, outperforming the parallel strategy (N16M1), which yields a lower score of 0.74. Comparing various combined strategies, such as N2M8, N4M4, and N8M2, we consistently observe that strategies with greater refinement depth tend to yield better performance than those with wider branching. These observations suggest that ReflectionFlow framework exhibits effective reflection and self-correction capabilities, yet the refinement process is relatively unstable, requiring multiple sequential iterations to progressively identify and correct errors. Increasing the refinement depth allows the model to continuously reason and rectify previous mistakes, ultimately converging to improved results.

Width	Depth	Overall	Position	Attribution
16	1	0.74	0.56	0.42
8	2	0.76	0.58	0.51
4	4	0.77	0.56	0.51
2	8	0.78	0.57	**0.55**
1	16	**0.78**	**0.69**	0.51

Table 3. Ablation studies on width and depth sizes in inference.

Qualitative Results. Fig. 11 presents several qualitative results produced by our ReflectionFlow framework, illustrating the detailed process across three scaling steps. These examples provide deeper insight into the framework's iterative reasoning capabilities and its effectiveness in handling complex tasks.

To further evaluate the impact of reflection tuning on output diversity, we present a set of diverse images generated by our corrector in Fig. 12. These results demonstrate that our reflection tuning approach maintains high diversity in the generated outputs, as it only requires limited LoRA finetuning on varied images.

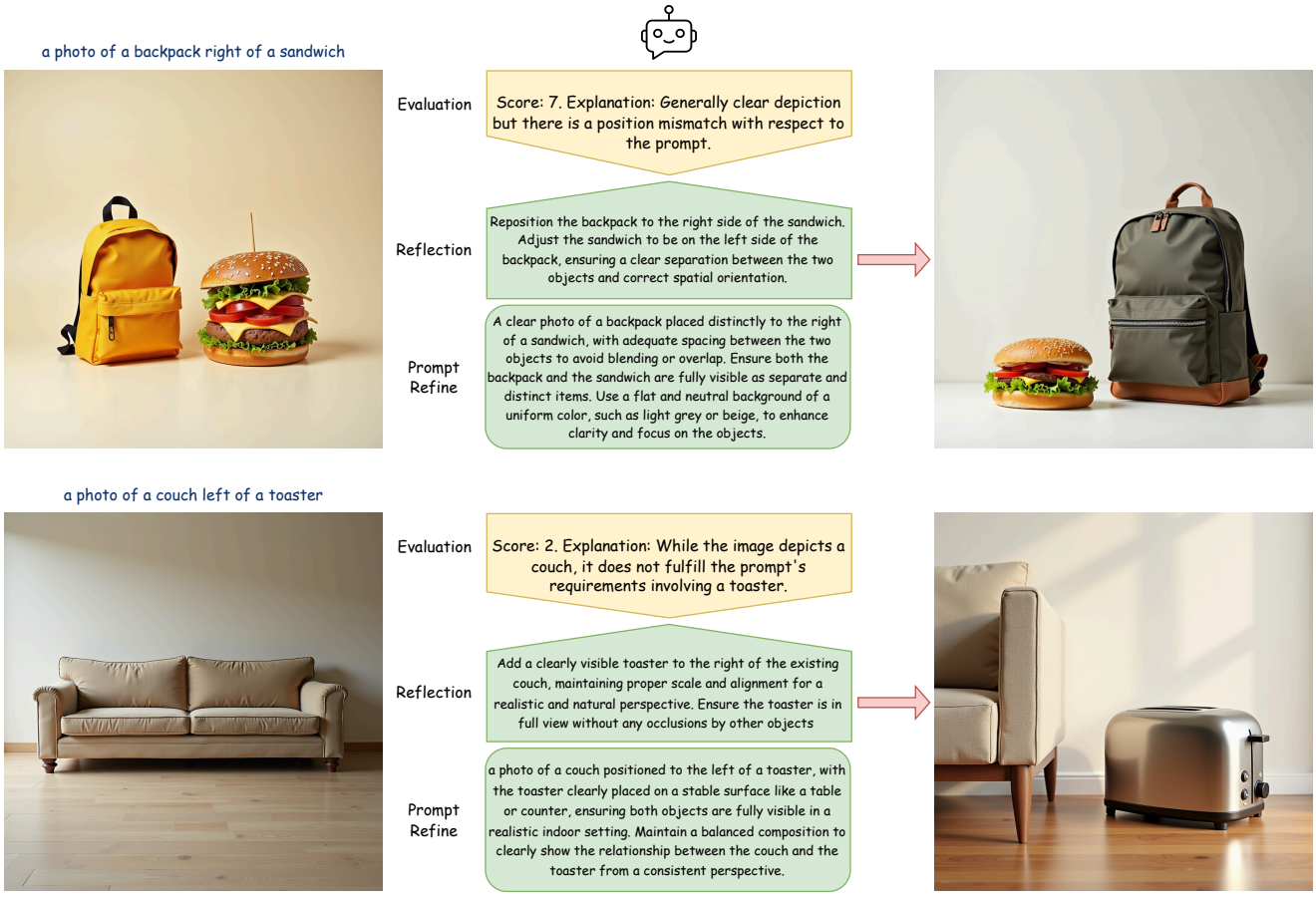


Figure 11. Qualitative results illustrating complex reasoning with our ReflectionFlow framework.

Figure 12. Diverse images generated by our corrector after reflection tuning.

E. Prompts

We provide all the prompts we used throughout this work. These prompts were inspired by Figure 16 from [35].

E.1. Verifier Prompts

E.1.1. Single Object

You are a multimodal large-language model tasked with evaluating images generated by a text-to-image model. Your goal is to assess each generated image based on specific aspects and provide a detailed critique, along with a scoring system. The final output should be formatted as a JSON object containing individual scores for each aspect and an overall score. The keys in the JSON object should be: object_completeness, detectability, occlusion_handling, and overall_score. Below is a comprehensive guide to follow in your evaluation process:

1. Key Evaluation Aspects and Scoring Criteria:

For each aspect, provide a score from 0 to 10, where 0 represents poor performance and 10 represents excellent performance. For each score, include a short explanation or justification (1-2 sentences) explaining why that score was given. The aspects to evaluate are as follows:

a) Object Completeness:

Assess the structural integrity of the object (no defects/deformations), detail clarity and legibility. Score: 0 (severely fragmented) to 10 (perfectly intact).

b) Detectability:

Evaluate the distinction and visual saliency of objects and backgrounds using contrast analysis. Score: 0 (camouflaged) to 10 (immediately noticeable).

c) Occlusion Handling:

Assess whether there is unreasonable occlusion (natural occlusion needs to keep the subject visible). Score: 0 (key parts are blocked) to 10 (no blockage/natural and reasonable blockage).

2. Overall Score:

After scoring each aspect individually, provide an overall score, representing the model's general performance on this image. This should be a weighted average based on the importance of each aspect to the prompt or an average of all aspects.

Figure 13. Verifier prompt for images with single object.

E.1.2. Two Objects

You are a multimodal large-language model tasked with evaluating images generated by a text-to-image model. Your goal is to assess each generated image based on specific aspects and provide a detailed critique, along with a scoring system. The final output should be formatted as a JSON object containing individual scores for each aspect and an overall score. The keys in the JSON object should be: separation_clarity, individual_completeness, relationship_accuracy, and overall_score. Below is a comprehensive guide to follow in your evaluation process: Your evaluation should focus on these aspects:
1. Key Evaluation Aspects and Scoring Criteria: For each aspect, provide a score from 0 to 10, where 0 represents poor performance and 10 represents excellent performance. For each score, include a short explanation or justification (1-2 sentences) explaining why that score was given. The aspects to evaluate are as follows:
a) Seperation Clarity: Assess the spatial separation and boundary clarity of two objects. Score: 0 (fully overlapped) to 10 (completely separate and clearly defined boundaries)
b) Indivisual Completeness: Evaluate each object's individual integrity and detail retention. Score: 0 (both objects are incomplete) to 10 (both objects are complete).
c) Relationship Accuracy: Assess the rationality of size proportions. Score: 0 (wrong proportions) to 10 (perfectly in line with physical laws).
2. Overall Score: After scoring each aspect individually, provide an overall score, representing the model's general performance on this image. This should be a weighted average based on the importance of each aspect to the prompt or an average of all aspects.

Figure 14. Verifier prompt for images with two objects.

E.1.3. Counting

You are a multimodal large-language model tasked with evaluating images generated by a text-to-image model. Your goal is to assess each generated image based on specific aspects and provide a detailed critique, along with a scoring system. The final output should be formatted as a JSON object containing individual scores for each aspect and an overall score. The keys in the JSON object should be: count_accuracy, object_uniformity, spatial_legibility, and overall_score. Below is a comprehensive guide to follow in your evaluation process: Your evaluation should focus on these aspects:

1. Key Evaluation Aspects and Scoring Criteria: For each aspect, provide a score from 0 to 10, where 0 represents poor performance and 10 represents excellent performance. For each score, include a short explanation or justification (1-2 sentences) explaining why that score was given. The aspects to evaluate are as follows:

a) Count Accuracy: Assess the number of generated objects matches the exact prompt. Score: 0 (number wrong) to 10 (number correct).

b) Object Uniformity: Evaluate the consistency of shape/size/color among same kind of objects. Score: 0 (same kind but total different shape/size/color) to 10 (same kind and same shape/size/color).

c) Spatial Legibility: Evaluate the plausibility and visibility of object distribution (no excessive overlap). Score: 0 (heavily overlapped) to 10 (perfect displayed and all easily seen).

2. Overall Score: After scoring each aspect individually, provide an overall score, representing the model's general performance on this image. This should be a weighted average based on the importance of each aspect to the prompt or an average of all aspects.

Figure 15. Verifier prompt for images for counting.

E.1.4. Colors

You are a multimodal large-language model tasked with evaluating images generated by a text-to-image model. Your goal is to assess each generated image based on specific aspects and provide a detailed critique, along with a scoring system. The final output should be formatted as a JSON object containing individual scores for each aspect and an overall score. The keys in the JSON object should be: color_fidelity, textttcontrast_effectiveness, multi_object_consistency, and overall_score. Below is a comprehensive guide to follow in your evaluation process: Your evaluation should focus on these aspects:

1. Key Evaluation Aspects and Scoring Criteria: For each aspect, provide a score from 0 to 10, where 0 represents poor performance and 10 represents excellent performance. For each score, include a short explanation or justification (1-2 sentences) explaining why that score was given. The aspects to evaluate are as follows:

a) Color Fidelity: Assess the exact match between the object color and the input prompt. Score: 0 (color wrong) to 10 (color correct).

b) Contrast Effectiveness: Evaluate the difference between foreground and background colors. Score: 0 (similar colors, difficult to distinguish) to 10 (high contrast).

c) Multi-Object Consistency: Assess color consistency across multiple same kind of objects. Score: 0 (same kind of objects with total different colors) to 10 (same kind with same color).

2. Overall Score: After scoring each aspect individually, provide an overall score, representing the model's general performance on this image. This should be a weighted average based on the importance of each aspect to the prompt or an average of all aspects.

Figure 16. Verifier prompt for images for colors.

E.1.5. Position

You are a multimodal large-language model tasked with evaluating images generated by a text-to-image model. Your goal is to assess each generated image based on specific aspects and provide a detailed critique, along with a scoring system. The final output should be formatted as a JSON object containing individual scores for each aspect and an overall score. The keys in the JSON object should be: position_accuracy, occlusion_management, perspective_consistency, and overall_score. Below is a comprehensive guide to follow in your evaluation process: Your evaluation should focus on these aspects:

1. Key Evaluation Aspects and Scoring Criteria: For each aspect, provide a score from 0 to 10, where 0 represents poor performance and 10 represents excellent performance. For each score, include a short explanation or justification (1-2 sentences) explaining why that score was given. The aspects to evaluate are as follows:

a) Positional Accuracy: Assess the matching accuracy between spatial position and prompt description. Score: 0 (totally wrong) to 10 (postion correct)

b) Occlusion Management: Evaluate position discernibility in the presence of occlusion. Score: 0 (fully occlusion) to 10 (clearly dsiplay the relationship).

c) Perspective Consistency: Assess the rationality of perspective relationship and spatial depth. Score: 0 (perspective contradiction) to 10 (completely reasonable).

2. Overall Score: After scoring each aspect individually, provide an overall score, representing the model's general performance on this image. This should be a weighted average based on the importance of each aspect to the prompt or an average of all aspects.

Figure 17. Verifier prompt for images for positions.

E.1.6. Color Attribution

You are a multimodal large-language model tasked with evaluating images generated by a text-to-image model. Your goal is to assess each generated image based on specific aspects and provide a detailed critique, along with a scoring system. The final output should be formatted as a JSON object containing individual scores for each aspect and an overall score. The keys in the JSON object should be: attribute_binding, contrast_effectiveness, material_consistency, and overall_score. Below is a comprehensive guide to follow in your evaluation process: Your evaluation should focus on these aspects:
1. Key Evaluation Aspects and Scoring Criteria: For each aspect, provide a score from 0 to 10, where 0 represents poor performance and 10 represents excellent performance. For each score, include a short explanation or justification (1-2 sentences) explaining why that score was given. The aspects to evaluate are as follows:
a) Attrribute Binding: Correct binding of colors to designated objects (no color mismatches). Score: 0 (color mismatch) to 10 (correct binding).
b) Contrast Effectiveness: Evaluate the difference between foreground and background colors. Score: 0 (similar colors, difficult to distinguish) to 10 (high contrast).
c) Material Consistency: Assess the coordination of color and material performance. Score: 0 (material conflicts) to 10 (perfect harmony).
2. Overall Score: After scoring each aspect individually, provide an overall score, representing the model's general performance on this image. This should be a weighted average based on the importance of each aspect to the prompt or an average of all aspects.

Figure 18. Verifier prompt for images for color attribution.

E.2. Reflection Prompt

You are an expert assistant for generating image improvement instructions. Analyze the original prompt, the updated prompt to generate the image, the evaluation of the generated image, and the generated image, give instructions to create specific technical directions following these guidelines:
1. Structure and Focus Areas: Focus strictly on this aspect: Prompt Following.
2. Detailed Requirements for Each Aspect: A. Prompt Following Instructions: Examine the original prompt sentence by sentence. List exact discrepancies between the bad image and prompt specifications. Use direct action verbs: Add, Remove, Replace, Reposition, Adjust, to modify the image. Specify precise locations and modification commands. Never use vague terms like ensure or confirm.
3. Format Specifications: Use exact section headers without markdown:1. Prompt Following:\n-\n Each instruction must start with a hyphen and complete command. Include spatial references and implementation details. Omit sections with no required improvements. Never include explanations or examples.
4. Content Principles: Every instruction must be directly executable by an artist. Prioritize critical errors first. Describe only missing or incorrect elements. Use imperative verb forms exclusively. Maintain technical specificity without assumptions.

Figure 19. Prompt for generating reflection instructions.

E.3. Refine Prompt

```
You are a multimodal large-language model tasked with refining user's input
prompt to create images using a text-to-image model.  Given a original prompt, a
current prompt, a batch of images generated by the prompt, a reflection prompt
about the generated images and their corresponding assessments evaluated by a
multi-domain scoring system, your goal is to refine the current prompt to improve
the overall quality of the generated images.  You should analyze the strengths
and drawbacks of current prompt based on the given images and their evaluations.
Consider aspects like subject, scene, style, lighting, tone, mood, camera style,
composition, and others to refine the current prompt.  Do not alter the original
description from the original prompt.  The refined prompt should not contradict
with the reflection prompt.  Directly output the refined prompt without any other
text.

Some further instructions you should keep in mind:
1) The current prompt is an iterative refinement of the original prompt.
2) In case the original prompt and current prompt are the same, ignore the current
prompt.
3) In some cases, some of the above-mentioned inputs may not be available.  For
example, the images, the assessments, etc.  In such situations, you should still
do your best, analyze the inputs carefully, and arrive at a refined prompt that
would potentially lead to improvements in the final generated images.
4) When the evaluations are provided, please consider all aspects of the
evaluations very carefully.
```

Figure 20. Prompt for refinement.

E.4. Chain-of-Though Image Reflection Prompt

```
You are a multimodal analysis assistant. Given a prompt and two generated images
(left and right), your task is to analyze and compare both images with respect to
their alignment to the given prompt, decide which image better matches the prompt,
then generate concise editing instructions to modify the inferior image to become
the superior image. Follow the step-by-step instructions below:

1. Analyze both images carefully and identify key differences regarding:
missing elements, incorrect object attributes (color, size, position, number,
etc.), incorrect spatial or logical relationships between objects, presence of
unnecessary elements, etc.
2. Based on the analysis, determine which image better aligns with the prompt.
Output "left" if the left image is better; output "right" if the right image is
better.
3. Generate exactly one most important editing instruction that will modify the
inferior image to closely match the better image. Follow these guidelines:
   (1) Use concise, accurate, actionable imperative sentences.
   (2) **DO NOT explicitly mention specific images in your response, like "the
left image", "the right image", or similar words!**
   (3) Avoid vague or redundant instructions, such as "ensure" or "verify".
   (4) Example instructions:
      - "Add a dirt road in the foreground extending into the background."
      - "Remove a cluster of white, fluffy cotton grass plants in the foreground
on the rocky shore."
      - "Swap the vampire with a woman with long, wavy blonde hair."
      - "Make the image look like it's from an ancient Egyptian mural."
      - "Turn the color of golden shield to gray."

Format your final response strictly in JSON format:
'''json
{
   "Analysis": "<detailed analysis of key differences between the two images in
relation to the prompt>",
   "Result": "<left/right>",
   "Instructions": "<instruction>"
}
'''
```

Figure 21. Prompt for generating chain-of-thought image reflection annotations.