# A. Additional Results and Discussions

## A.1. Generalizability of Reflect-DiT

In this section, we provide additional discussion on DPG-Bench [18] results provided in Tab. 5 to highlight the effectiveness of Reflect-DiT on a broad range of prompts. We note that several frontier open-sourced models achieve similar performance on this benchmark, in the range of 83.0-85.0. The performance gap among models on DPG-Bench is less pronounced compared to GenEval. Analysis of SANA-1.0 outputs indicates that approximately 75% of the prompts are less challenging, as indicated by the base model achieving a score above 0.8 without inference-time scaling. This may explain why SANA-1.5 [61] reported inference-time scaling results only on GenEval. To further illustrate the effectiveness of Reflect-DiT, we construct two challenging datasets by subsampling prompts where the SANA-1.0-1.6B base model scores poorly in the single-sample setup. Specifically, we create a subset of 246 prompts (Hard-246) consisting of prompts on which the base model obtained a score below 0.8 in the single-sample setup, and a subset of 56 prompts (Hard-56) consisting of prompts on which the base model obtained a score below 0.5 in the single-sample setup. We compare with the base model and best-of-N sampling on the two subsets as well as the full benchmark. Reflect-DiT achieves better performance on all three datasets, with more pronounced differences on the two hard subsets. The results on DPG-Bench, together with the main results on GenEval and human evaluations on PartiPrompts, demonstrate the effectiveness of Reflect-DiT across diverse text-to-image generation tasks.

## A.2. Inference Speed

We benchmarked the inference speed of Reflect-DiT against the best-of-N baseline and found no significant difference in performance. Overall, Reflect-DiT and best-of-N sampling achieved similar throughput: 11.32 samples per minute for Reflect-DiT and 10.12 samples per minute for best-of-N, where each sample includes a generated image and corresponding text feedback. Conceptually, generating N samples using Reflect-DiT has a similar computational cost to the best-of-N baseline, as both involve generating N images and running the VLM model N times. The only extra overhead comes from (1) encoding the images and text in the context and (2) computing cross-attention with extra keys and values. Furthermore, step (1) can be amortized across the denoising steps, as the context needs to be encoded only once per generated image at the beginning of the denoising loop. For N=20, the end-to-end latency is 118.57 seconds for Reflect-DiT and 105.98 seconds for the best-of-N baseline. Of the total time, 75.5% is spent generating images with the DiT and 24.5% is used for VLM inference.

## A.3. Scaling

We conducted multiple experiments with varying number of Transformer layers in the Context Transformer (Tab. 6). Results show that increasing the number of Transformer layers leads to an improvement in model performance.

| | Number of Samples | | | | |
|---|---|---|---|---|---|
| | 4 | 8 | 12 | 16 | 20 |
| Baseline | 0.705 | 0.731 | 0.743 | 0.749 | 0.751 |
| +1 Layer | 0.769 | 0.790 | 0.797 | **0.801** | 0.804 |
| +2 Layers | **0.776** | **0.792** | **0.799** | 0.799 | **0.807** |

Table 6. **Ablation study on number of Transformer layers.**

## A.4. Connection with Instruction-Guided Image Editing

Several works have focused on instruction-following image editing models, *e.g.* InstructPix2Pix [4] and InstructDiffusion [13]. These models take an input image and natural language instruction and perform the desired edit on the image. Compared to these methods, Reflect-DiT has two key advantages. First, our training data relaxes the strict requirement for paired input-edited images. Image-editing data consists of paired images: an input and a corresponding edited version that adheres to the instruction while maintaining visual consistency.

Reflect-DiT only requires a "good" image that avoids a problem found in a "bad" image, which can be easily collected. Second, but more importantly, Reflect-DiT uniquely leverages multi-round feedback context and progressively refines its generations. Our results in Section 4.7.1 demonstrate that iterative in-context feedback significantly improves performance. Figure 6 also illustrates that multi-round feedback enables the progressive refinement of generations. Although some identified defects may not be fully resolved in a single iteration, the generated images converge towards a correct image after multiple rounds. The model learns this ability despite the random sampling of in-context feedback during training.

## A.5. Connection with Self-Correcting T2I Agent

Prior to the recent interest in inference-time scaling, several works attempted to achieve self-verification and correction through an agentic framework, such as SLD [58] and GenArtist [53]. These works employ a frontier LLM or VLM (e.g. GPT4) to control a set of external tools such as object detectors, segmentation models and inpainting models to verify the accuracy of text-to-image (T2I) generation and apply corrective operations to the generated image. These approaches suffer from key scalability and flexibility issues.

In terms of scalability, calling proprietary APIs for each inference is expensive. Additionally, generating function calls auto-regressively and executing multiple models per refinement round introduce significant computational overhead and latency. In contrast, Reflect-DiT and recent works on inference-time scaling only require a, significantly smaller, VLM judge model to simply generate concise per-image feedback in natural language.

In terms of flexibility, the success of these agentic frameworks depends on all submodules executing successfully, giving rise to two main problems. First, these submodules may not be up-to-date. For example, the inpainting and image editing models they use are primarily based on SDv1.5 [43] or SDXL [39], resulting in suboptimal generation quality. Updating all tools to the latest architectures and base models is non-trivial, since the developers of these tools may discontinue maintenance, which is not unrealistic for most research projects. Additionally, adapting a system with numerous components to custom use cases can be challenging. For example, if a user wants to generate a painting, the pretrained object detector and segmentation models may fail on out-of-distribution cases such as painting generation. Collecting a detection and segmentation dataset and fine-tuning the object detector and segmentation model can be expensive and challenging, not to mention the difficulty of data collection for others tools like inpainting and image-editing models. In contrast, Reflect-DiT and recent inference-time scaling methods can easily adapt to new use cases as long as a judge model provides feedback, which can be obtained by fine-tuning a strong foundational VLM on limited data. In fact, we show in Figure 6 (main paper) that Reflect-DiT can adapt to novel use cases such as painting generation in a zero-shot manner due to the inherent generalizability of VLMs, highlighting the flexibility of our approach.

We find these works interesting but tangential. In our early explorations, we attempted to reproduce the findings of GenArtist [53] (NeurIPS 2024) and test its performance on the GenEval benchmark. Unfortunately, the model fails to complete the official demo script using the default prompt, as the latest version of GPT-4 generates ill-formed function calls approximately 20 seconds into the agentic loop. Our experience further highlights the inflexibility and inconsistency of these methods.

For completeness, we report comparison against SLD[58], a LLM-based agent.

### A.6. Connection with Reinforcement Learning (RL)

If we consider the consecutive, non-i.i.d. generative process of multiple image samples as a policy optimization problem, then Reflect-DiT's training objective can be viewed as equivalent to imitation learning, where we directly apply the SFT objective to a set of target "good actions",*i.e.* accurate image generations. We also explored reinforcement learning objectives such as D3PO [64] and Diffusion-DPO [52], which incorporate negative samples during training. However, we encountered issues with training stability. We achieved state-of-the-art results using only the SFT objective and leave further exploration of RL objectives to future work. Our results mirror those of DeepSeek-R1 [15], where the authors showed that smaller LLMs can achieve substantial performance gains solely through SFT on high-quality reasoning trajectories generated by a larger model, without requiring reinforcement learning.

| | GenEval$\uparrow$ | HPSv2$\uparrow$ |
|---|---|---|
| SANA-1.0 | .62 | 38.7 |
| SLD (N=20) | .67 | 38.4 |
| Reflect-DiT (N=20) | **.81** | **39.2** |

Table 7. **Further comparisions on GenEval.** We report Gen-Eval scores and HPSv2 scores.

### A.7. Concurrent Works

Following the success of test-time scaling in the language domain, the community has shown growing interest in applying it to text-to-image generation. Concurrently with this work, SANA-1.5 [61] explored best-of-N sampling on a state-of-the-art DiT. Our proposed Reflect-DiT outperforms SANA-1.5, which uses 2048 samples (best-of-2048), with only 20 samples by leveraging a reflection mechanism. Also concurrent, FK-steering [47] proposed a novel latent-space search method that extends beyond random search. However, its implementation is limited to the DDIM sampler and is not easily adaptable to multi-step solvers such as the DPM-Solver++ which is used by SANA. In contrast, Reflect-DiT has a constant memory footprint, making it more scalable. After testing the official SDXL-based implementation, we find that FK-Steering causes out-of-memory errors at 20 particles. While their results are promising, we believe it has considerable room for improvement, particularly in adapting to state-of-the-art DiTs and optimizing memory usage. Another concurrent work [16] explored generating images using chain-of-thought (CoT) [54] reasoning and incorporates elements of self-verification and iterative improvement. However, their work focuses on autoregressive image generation models and is a direct adaptation of analogous approaches in the language domain.

## B. Limitations

While we have demonstrated Reflect-DiT's effectiveness across various applications, we acknowledge its limitations. First, the training data used for the VLM judge primarily focuses on prompt alignment, e.g. whether there are sufficient objects, whether they satisfy positional constraints,

**VLM Training Data Template**

```
{
"from":  "human",
"value":  "<image>\n
Please evaluate this generated
image based on the following
prompt:  [[prompt]].
Focus on text alignment and
compositionality."
},
{
"from":  "gpt",
"value":  "[[feedback_text]]"
}
```

Table 8. VLM Training Data Template

*etc*. Thus, the VLM judge may not be able to provide proper feedback or suggest improvements for other aspects, such as the aesthetics of an image. In general, natural language feedback datasets of this kind are more difficult to collect. We hope future datasets can help address this issue. Second, we observe that the VLM judge suffers from hallucinations, similar to its base model. We present examples of these errors in Figure 7. For example, in row 3 of Figure 7, the VLM mistakenly claims that the boat is not present in the image, despite the boat being clearly present and the image being correct. Lastly, we observed that the diffusion model may fail to address certain forms of feedback in a single iteration. In some cases, we observe the model iteratively refining the generation toward correctness, though it takes multiple iterations for the image to become fully aligned with the prompt. For example, in Figure 6 (main paper), row 3, we observe that the position of the dog and the tie gradually move toward the desired layout. In other cases, the progression is less interpretable. For example, in Figure 6 (main paper), row 2, the generated image should contain three seeds, but it undergoes an inconsistent progression of 2-5-1-3. Empirically, we observe that Reflect-DiT can generate accurate, text-aligned images with fewer inference-time samples, achieving a 22% improvement on the counting subcategory of GenEval (Table 1 in main paper).

# C. Technical Details

## C.1. VLM Training

Following SANA-1.5 [61], we format the VLM training data into a conversation format. Our template differs from SANA because we use a different base model, Qwen-2.5-VL 3B [2]. We present the template in Table 8. We provide hyperparameters of our training run in Table 9.

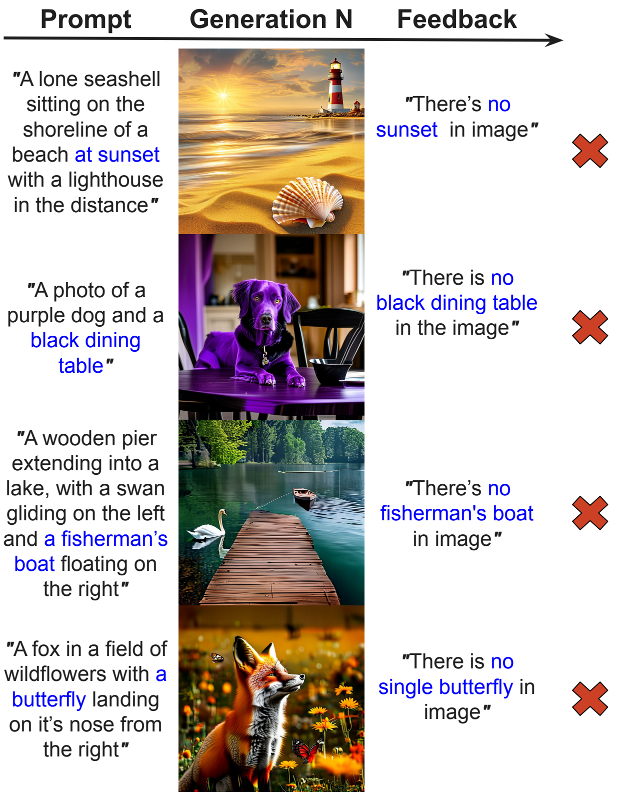

Prompt
Generation N
Feedback
"A lone seashell sitting on the shoreline of a beach at sunset with a lighthouse in the distance"
"There's no sunset in image"
✗
"A photo of a purple dog and a black dining table"
"There is no black dining table in the image"
✗
"A wooden pier extending into a lake, with a swan gliding on the left and a fisherman's boat floating on the right"
"There's no fisherman's boat in image"
✗
"A fox in a field of wildflowers with a butterfly landing on it's nose from the right"
"There is no single butterfly in image"
✗

Figure 7. **Failure cases of Reflect-DiT**. Failure cases of Reflect-DiT. While Reflect-DiT demonstrates strong refinement capabilities, the generated feedback can occasionally introduce errors between iterations. In the first example, the model fails to recognize that the specific lighting conditions signify a "sunset", leading to an incorrect adjustment. Similarly, in the second example, the model struggles to distinguish the color of the "dining table" because the purple hue from the "dog" reflects off the table, creating ambiguity. These cases highlight subjectivity in the VLM evaluation, where the model's interpretation may still be reasonable. However, the final two examples illustrate more typical failure cases. In both images, objects ("boat" and "butterfly") are completely overlooked by the feedback model. This issue likely arises because the objects are too small or unusually shaped, which makes them difficult to detect, resulting in incorrect evaluations.

## C.2. Diffusion Transformer

### C.2.1. Vision-Encoder

The vision encoder is a SigLIP-Large [67] that encodes each image into a feature map of size $24\times24=1024$.The feature map is then downsampled to $8\times8=64$ via average pooling and flattened into a 1D sequence of length 64. We then use a two layer MLP with GELU activation to project the features to match the input dimension of the Context Transformer. To improve training stability, we add an RMSNorm layer after the projector. Before training, we freeze the SigLIP model. The projector is trained end-to-end with the rest of

| Hyperparameters | VLM Judge | Reflect-DiT | SFT | Diffusion-DPO |
|---|---|---|---|---|
| Learning Rate | 1e-5 | 1e-5 | 1e-5 | 1e-5 |
| Batch Size | 48 | 48 | 48 | (24, 24)* |
| Weight Decay | 0.1 | 0 | 0 | 0 |
| Optimizer | AdamW | CAME | CAME | CAME |
| Schedule | 1 epoch | 5k step | 5k step | 5k step |
| Warmup steps | 0.03 epoch | 500 step | 500 step | 500 step |

Table 9. **Hyperparameters used for each experiment.** * We use 24 positive samples and 24 negative samples per batch.

the DiT.

### C.2.2. Text-Encoder

We use Gemma-2B [49] as the text encoder for text feedback. It is kept frozen during training. Since Gemma-2B is also used by SANA as the prompt encoder, no additional parameters are introduced to the overall system.

### C.2.3. Context Transformer

The Context Transformer is a two-layer Transformer. Its primary purpose is to (1) align encoded features with the features space of the base DiT and (2) associate the feedback with the corresponding image. Each Context Transformer consists of a standard Transformer block, including a self-attention layer followed by a feed-forward network. We use the exact FFN design of Qwen2.5-VL [2]. For the self-attention layer, we incorporated rotary positional embeddings [17] following the design of many modern LLMs and VLMs.

### C.2.4. Training

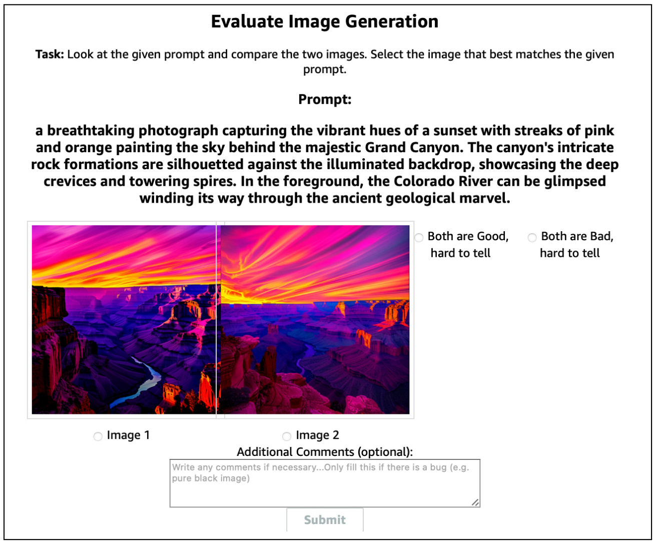
We report the training hyperparameters for Reflect-DiT, SFT, and Diffusion-DPO baselines in Table 9. We use the CAME optimizer [32] to train the DiT, following the approach in SANA [61]. For Diffusion-DPO, we tested three values of $\beta$, the hyperparameter controlling the KL divergence penalty, and determined that $\beta = 2000$ produces the optimal result.

## C.3. Human Evaluation Details

We use Amazon Mechanical Turk for human evaluations. We present the user interface provided to human annotators in Figure 8. We collect three evaluations per image pair and compared Reflect-DiT(N=20) with best-of-20 for each prompt. We randomly selected 100 prompts from the PartiPrompts dataset and generated 100 corresponding image pairs for human annotators. In total, 300 annotations were collected.

## D. Additional Qualitative Examples

We present additional evaluation results in Figure 9. Examples 1 and 6 demonstrate how Reflect-DiT guides the generation process to accurately position objects within a scene. Examples 2, 4, and 7 focus on object counting, ensuring that the correct number of distinct items. Example 3 presents a particularly complex prompt, where Reflect-DiT accurately positions all objects while maintaining the correct quantity, such as the specified number of “wooden barrels”. Lastly, Example 5 highlights a challenging case—separating object identity from color attributes—that many generative models struggle with. Typically, models often conflate color and object identity, making requests like ”a black sandwich” difficult to fulfill. However, Reflect-DiT successfully distinguishes these attributes, demonstrating its advanced capability to handle nuanced prompts.



**Evaluate Image Generation**

**Task:** Look at the given prompt and compare the two images. Select the image that best matches the given prompt.

**Prompt:**

**a breathtaking photograph capturing the vibrant hues of a sunset with streaks of pink and orange painting the sky behind the majestic Grand Canyon. The canyon's intricate rock formations are silhouetted against the illuminated backdrop, showcasing the deep crevices and towering spires. In the foreground, the Colorado River can be glimpsed winding its way through the ancient geological marvel.**

○ Both are Good, hard to tell

○ Both are Bad, hard to tell

○ Image 1

○ Image 2

Additional Comments (optional):

Write any comments if necessary...Only fill this if there is a bug (e.g. pure black image)

Submit

Figure 8. **User interface for human annotators.**

## E. Reproducibility Statement

We will release the training code and data for the DiT and VLM judge model, and pretrained checkpoints. We will also release the generated images that produce the main result on GenEval benchmark. Additionally, we will release the list of prompts in Hard-246 and Hard-56 subset of DPG-Bench.

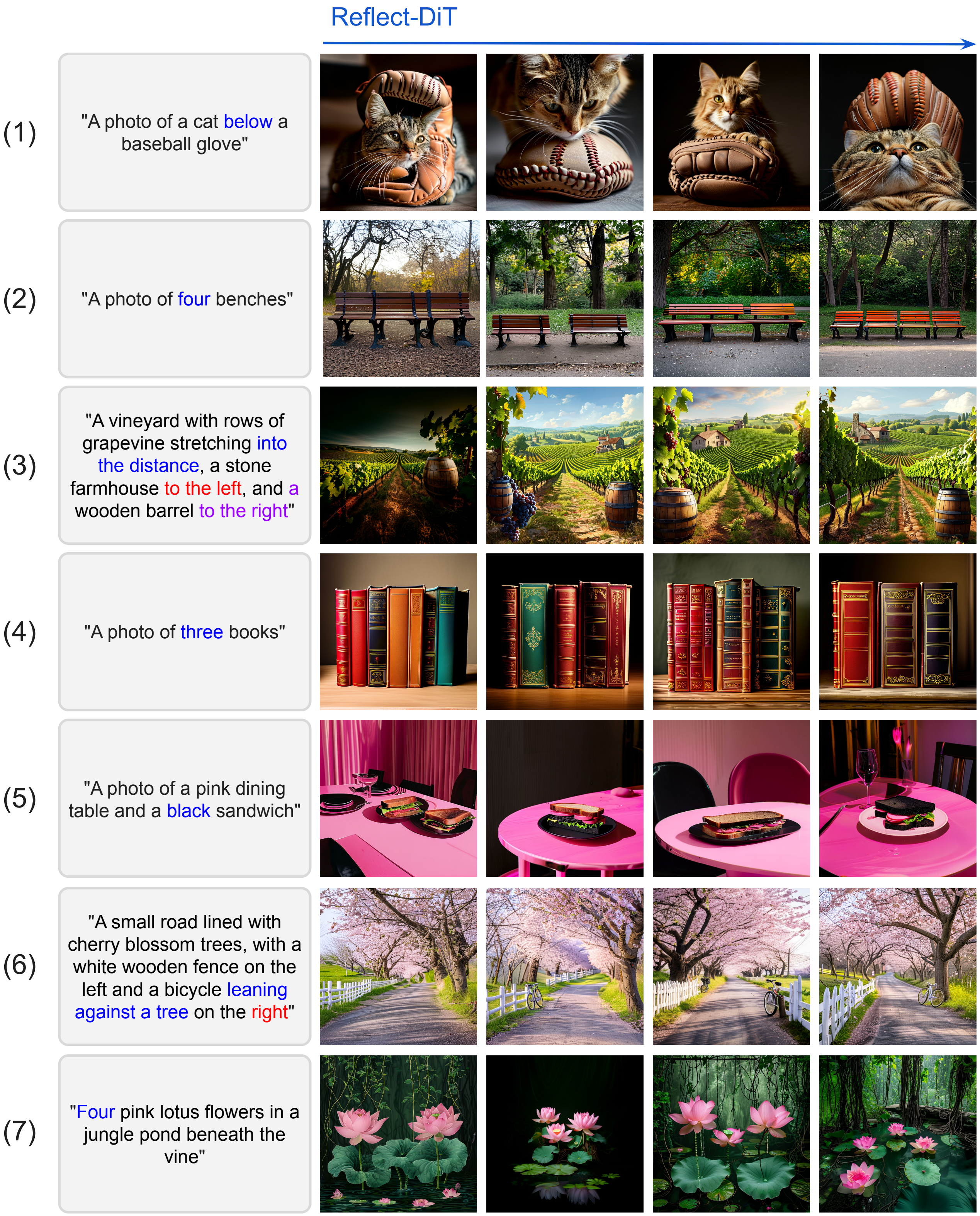

Reflect-DiT
(1)
"A photo of a cat below a baseball glove"
(2)
"A photo of four benches"
(3)
"A vineyard with rows of grapevine stretching into the distance, a stone farmhouse to the left, and a wooden barrel to the right"
(4)
"A photo of three books"
(5)
"A photo of a pink dining table and a black sandwich"
(6)
"A small road lined with cherry blossom trees, with a white wooden fence on the left and a bicycle leaning against a tree on the right"
(7)
"Four pink lotus flowers in a jungle pond beneath the vine"

Figure 9. **Additional qualitative examples from Reflect-DiT.**