Appendix

1. Data Construction

1.1. Supervised Training

As illustrated in Fig. 1, we provide an overview of the data distribution utilized for supervised training across different tasks. For the **text-to-image generation** task, we employ a diverse set of datasets, including ShareGPT-4o-Image [3], SFHQ [1], FLUX-Reason-6M [7], comprising a total of 51M samples. For the **text rendering** task, we utilize DenseFusion [21] and internally collected text-containing data, resulting in 3M samples. The **image editing** task leverages UltraEdit [26], OmniConsistency [18], Echo4o [25], GPT-Image-Edit [22], ShareGPT-4o-Image [3], X2Edit [13], NHR [10], accumulating to 5M samples. For **in-context generation**, we use Nano-banana-150k[1] and Echo-4o-Image [25], totaling 200K samples.

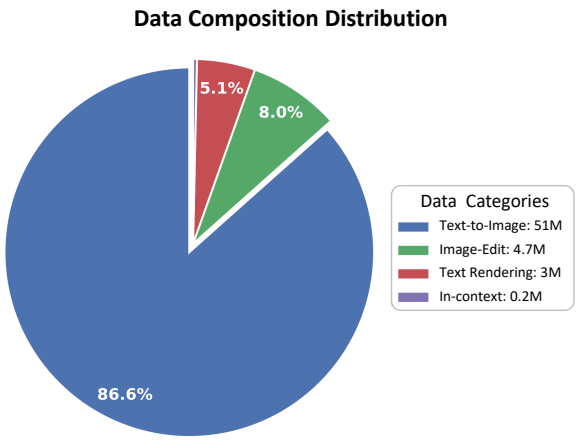

Figure 1. Data distribution of supervised training.

1.2. SepGRPO

Semantic Composition Dataset. We employ the Geneval-style training dataset from Flow-GRPO [12] as our semantic composition dataset. This dataset comprises prompts that specify object count, color, and relative spatial relationships, making it well-suited for training models to improve semantic alignment between generated images and textual descriptions.

Reasoning Generation Dataset. We collect 10K `prompt-prompt_rewrite` reasoning data pairs. In each pair, the `prompt` is intentionally ambiguous and necessitates world knowledge reasoning for text-to-image (T2I) generation, whereas the corresponding `prompt_rewrite` is explicit and can be directly used for T2I image generation without further reasoning. Specifically, we incorporate six types of world knowledge and their respective sub-categories, consistent with the WISE benchmark [15]. For each sub-category, we employ GPT [16] to construct `prompt-prompt_rewrite` pairs (Tab. 1). To ensure the uniqueness, we apply SequenceMatcher for rigorous deduplication, guaranteeing no overlap between our synthesized pairs and the official WISE benchmark.

Text Rendering Dataset. We sample 3,000 captions from DataComp-1B [8] and employ Qwen3-32B [24] to rewrite these captions. This rewriting process augments the original descriptions by inserting contextually appropriate text onto specified objects (e.g., placing the word "coffee" on a cup). As a result, the captions are enriched with renderable textual content, making them well-suited for training SepGRPO.

Image Editing Dataset. We construct our image editing dataset by filtering 3,000 Pico-Banana-400K [17] samples with near-square aspect ratios (between 0.95 and 1.05). Since both source and target images are resized to square shapes during the MLLM-GRPO training stage, selecting near-square samples helps to minimize distortion caused by resizing. This preprocessing step also facilitates efficient, parallelized reward computation using the SigLIP-2 [20].

Reflection Dataset. We collected 3,000 reflection samples from GenRef-wds [27], a dataset specifically designed for reflection-based image generation. To ensure consistency between images before and after reflection, we exclusively used the *edit* subset in GenRef-wds.

2. Implementation Details

ThinkGen integrates Qwen3-VL-8B-Think [19] with OmniGen2-DiT-4B [23]. The connector is implemented as a simple linear layer that maps the hidden states from the last two layers of Qwen3-VL-8B-Think, reducing their dimensionality from 8,192 to 2,520 to match the input requirements of DiT. For the Prepadding States, we set K=25.

As shown in Tab. 2, we adopt a multi-stage supervised training strategy using a dynamic mixture of the curated data described in Sec. 1. Specifically, an alignment stage (Stage1) for initializing the connector, a large-scale pre-training stage (Stage2), and a supervised fine-tuning stage (Stage3) for high-quality fine-tuning.

During the SepGRPO phase, images are generated at a resolution 512×512 over 20 steps. The `cfg` parameter is set to 4 and is enabled only during the first 60% of steps to accelerate generation. The rollout parameters N_1 and N_2 are set to 8 and 24, respectively. In the DiT-GRPO stage, the loss is backward only for the first 60% of steps.

[1] 1. https://github.com/yejy53/Nano-banana-150k

###[System Role Instruction]
You are a prompt engineering expert.

###[User Input]
Please generate two prompts for AI image generation. These two prompts must incorporate sub-category in category knowledge.
- The first prompt (prompt1) is a more vague prompt that requires sub-category knowledge (this prompt should be as vague as possible and the sentence length should be as short as possible, less than 10 words). For specific writing methods, you can refer to **prompt1** in the Example.
- The second prompt (prompt_rewrite) should provide a straightforward, concrete description of the desired image. This prompt is a clear text-to-image prompt, which can be used to generate images for the text-to-image model without logical reasoning. This T2I_prompt should be as clear as possible. For specific writing methods, you can refer to **prompt_rewrite** in the Example)
**Return the output as Do not output anything else.

Output only a JSON list, no extra explanation. Strictly generate a list of 5 samples, nothing else. Each sample is a dictionary containing the two keys: "prompt1", and "prompt_rewrite".**

Table 1. **The template to generate reasoning data pairs.**

	Stage1	**Stage2**	**Stage3**
Learning Rate	1.0×10^{-3}	2.5×10^{-4}	1.0×10^{-4}
Batch Size	512	1280	64
LR scheduler	Cosine	Constant	Constant
Weight decay	0.0	0.0	0.0
Gradient Clip	1.0	1.0	1.0
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.95, \epsilon = 10^{-9}$)		
Warm-up steps	500	0	0
Training steps	47K	100K	11k
Drop Rate	10%	10%	0.01%
Data Size	24M	60M	0.7M
Gen resolution	512×512	512×512	1024×1024

Table 2. Implementation Details of ThinkGen.

3. SepGRPO Training Details

Input Format. During Supervised Pre-training and SepGRPO, we employ distinct data templates for generating pseudo-CoT annotations and for guiding the MLLM in CoT reasoning, as detailed in Sec. 4.1 and Sec. 4.2. Despite their differences, both templates share a common system prompt [SYS] (Tab. 3), which facilitating a cold start in the RL stages, and encouraging the MLLM to rewrite user input instructions favored by DiT.

Rule Models. SepGRPO employs distinct rule models tailored to each task, as detailed below:

- **Semantic Composition.** We use GenEval [9] to evaluate the consistency between generated images and provided instructions.
- **Reasoning generation:** For this task, images are generated from the prompt in our collected reasoning dataset. The generated image and its corresponding prompt_rewrite are then scored using HPSv3 [14].
- **Text rendering:** We utilize 3K prompts containing text rendering. The generated images are processed with OCR [5] to extract contained words, and generation quality is assessed via word accuracy [6].
- **Image editing:** 3K editing samples [17] are used for CoT reasoning editing. Both the generated images and ground truth are resized to 512×512, features are extracted using SigLIP2 [20], and editing quality is measured by cosine similarity.
- **Reflection:** For this task, 3K reflection samples are split evenly into prompt–bad_image and prompt–good_image pairs. The prompt–bad_image pairs use the corresponding editing instruction as ground-truth, while prompt–good_image pairs use "*The generated image is well aligned with the caption.*" as ground-truth. The Normalized Edit Distance (NED) is used to evaluate the MLLM's output. DiT is not used for this evaluation.

4. Supplemental Ablation Study

In this section, we present ablation studies on connector design and the extraction strategy for the </think> state to validate the effectiveness of our model architecture.

Connector Design. Tab. 4 compares the Stage1 results using different connector designs: a linear layer, an MLP, and a causal-transformer [11]. The results indicate that the simple linear layer achieves the best performance, outperforming more complex connectors such as MLP and causal-

```
###[System Role Instruction]
You are a helpful, general-purpose AI assistant with the ability to generate images and understand images.
Your primary goal is to assist the user effectively. When generating an image, provide a clear, one-sentence caption that accurately describes the requested image.

###[User Input]
caption or reference images + edit instruction
```

Table 3. **[SYS] for CoT reasoning.**

transformer.

Training stage	GenEval	WISE	ImgEdit
Linear (default)	0.78	0.46	3.93
MLP	0.73	0.43	3.78
Transformer	0.80	0.44	3.8

Table 4. Stage1 results of different connector designs. We use GenEval, WISE, ImgEdit for analysis.

Extraction strategy for the </think> state. In VGI-refine block, we truncate the hidden states preceding the </think> token, feeding only the subsequent hidden states into the DiT. As shown in Tab. 5, this strategy yields consistent improvements across all benchmarks, particularly for *short-prompt* generation tasks (GenEval: +0.12, WISE: +0.15, CVTG: +0.10, ImgEdit: +0.50). These results indicate that truncating the pre-</think> hidden states effectively eliminates redundant information, thereby enhancing image generation quality.

	Short-Prompt				*Long-Prompt*
	GenEval	WISE	CVTG	ImgEdit	DPG
CUT	0.78	0.46	0.28	3.93	80.86
ALL	0.66	0.31	0.18	3.43	80.60

Table 5. We analyze the impact of the extraction strategy in VGI-refine using GenEval, WISE, and CVTG. The CUT denotes using only the hidden states following the </think> token for image generation, while ALL employs all hidden states.

Ablation Study for Hidden State Layers. Tab. 6 compares different hidden states. Due to time constraints, we trained DiT on a representative data subset (150K samples from BLIP3o-60k [4] and ShareGPT [2]) instead of the full five-stage pipeline. The results indicate that utilizing the last two layers yields optimal performance. This finding aligns with conclusions from UniWorld [11].

Necessity of Multi-Stage Training. We investigate whether the proposed multi-stage training (Stages 1-3) can be replaced by simpler in Tab. 7. We first evaluate *Direct Inference*, which uses an off-the-shelf MLLM (Qwen3-VL-8B-Thinking) to generate a refined prompt for a standard OmniGen2. While feasible, it incurs computational overhead by requiring separate MLLM forward passes for prompt rewriting and hidden state extraction. Our integrated approach resolves this by yielding both the refined prompt and text hidden states in **a single CoT pass**. Furthermore, we evaluate a *Simplified Training* baseline that applies GRPO solely to the OmniGen2. This approach fails to improve reasoning performance (WISE score remains at 0.63), demonstrating that skipping the multi-stage alignment prevents the MLLM from enhancing its CoT reasoning capabilities. Thus, the heavy architectural coupling is necessary for both computational efficiency and effective reasoning optimization.

	GenEval	WISE	ImgEdit
1	0.32	0.22	2.59
2 (default)	0.42	0.25	2.89
4	0.43	0.26	2.83

Table 6. Ablation Study for Hidden State Layers.

	Params.	mllm forward	WISE	CVTG
ThinkGen (ours)	8B + 4B	1	0.76	0.84
OmniGen2 + Think	8B + 3B + 4B	2	0.63	0.51
OmniGen2 + Think †	8B + 3B + 4B	2	0.63	0.58

Table 7. † denote apply GRPO on Omnigen2 DiT.

Effectiveness of SepGRPO. We further ablate the necessity of our decoupled RL strategy (Stages 4 and 5). Note that Stage 4 specifically performs rollout on the MLLM to enhance CoT reasoning (WISE: 0.55→0.76), while Stage 5 focuses on the DiT for image quality (CVTG: 0.79→0.84). As shown in Tab. 8, *Skipping Stage 4* directly to Stage 5 fails to improve CoT quality, resulting in a significant drop in reasoning performance (WISE). Moreover, a *Joint Optimization* baseline—where both MLLM and DiT are unfrozen and trained simultaneously—proves detrimental. Joint RL destabilizes the MLLM's reasoning capabilities and breaks its pre-trained CoT structure. These findings validate that our sequential SepGRPO design is crucial for preventing optimization conflicts between the reasoning and

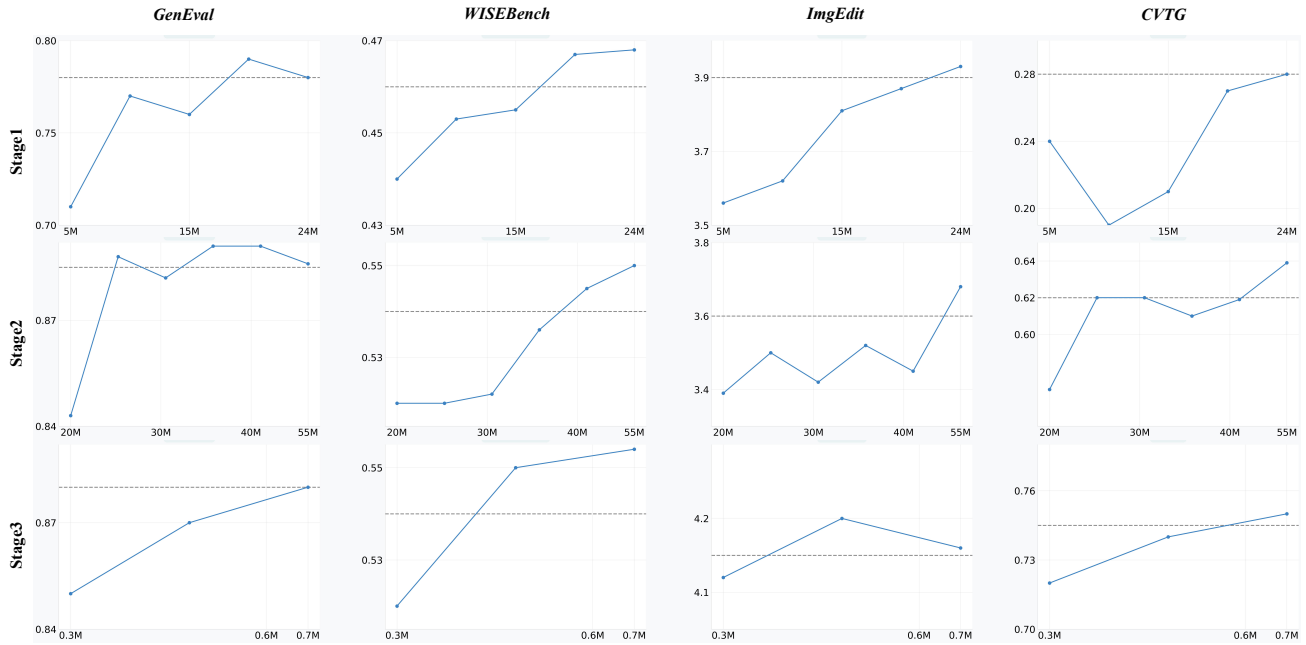

Figure 2. Data Scaling in Stage1-3.

generation modules.

	GenEval	WISE	ImgEdit
Stage4 & 5	0.89	0.76	4.21
Stage3→5	0.84	0.68	4.27
Stage3→5 †	0.82	0.67	3.99

Table 8. Ablation of SepGrpo. †: MLLM is trained.

Figure 3. Visualization of stage 1-5. Zoom in for a better view.

5. Data Scaling

We examine the scaling behavior of ThinkGen when increasing training data in Stage1-3 (Fig. 2). In Stage 2, our findings indicate that GenEval performance saturates at ∼88.5 when trained on 25M samples. WISE shows a steady improvement as the amount of training data increases. This suggests that the DiT's ability to follow simple instructions reaches its upper limit at 25M samples, while further increases in training data enable DiT to enhance its capability to perform reasoning generation.

6. Qualitative Analysis

To provide an interpretable understanding of our performance gains, we visualize the progressive improvements across the five training stages in Fig. 3.

- **Stages 1-3: Enhancing Instruction Following.** The initial stages focus on aligning the DiT with the MLLM's instructions. We observe a clear trajectory of correcting initial failure cases: early checkpoints struggle with precise control, whereas by Stage 3, the model's instruction-following capability is significantly enhanced, evidenced by progressively accurate text rendering and detail synthesis (Fig. 3, Row 2).
- **Stages 4-5: Boosting Reasoning and Aesthetics.** The subsequent RL stages further elevate the generation quality. Stage 4 (MLLM-GRPO) visually manifests the enhanced CoT reasoning capabilities (Row 1), ensuring that complex spatial relationships and logical constraints are accurately reflected in the generated images. Finally, Stage 5 (DiT-GRPO) optimizes the overall aesthetic quality and visual fidelity, resulting in highly coherent and visually pleasing outputs.
- **Extensive comparisons.** We provide a comparison of image generation and editing with thinking versus without in Fig. 4 and Fig. 5.

Prompt	Results	Results *w*/Thinking	
A caterpillar having completed its pupation process		<think> Okay, the user asked for an image of a caterpillar that has completed its pupation process. Let me break this down. First, I need to recall the life cycle of a butterfly or moth. Caterpillars go through metamorphosis, right? They start as eggs, then hatch into caterpillars, which eat and grow. Then they form a chrysalis or cocoon, which is the pupation stage. After that, they emerge as adult butterflies or moths. Actually, once the pupation is complete, the caterpillar is no longer a caterpillar. It's a butterfly or moth. So the user might have a misunderstanding. But the request is to generate an image of a caterpillar after pupation. Maybe they want the adult insect, but they called it a caterpillar by mistake. Alternatively, maybe they're referring to the chrysalis stage, but the user said "completed," so it's the end result. </think> Generate a realistic image of a newly emerged adult butterfly with delicate wings still slightly crumpled, standing on a leaf beside its empty chrysalis, showcasing vibrant wing patterns and fine details of its antennae and body.	
An animal, a symbol of strength in Brazil		**<think>** Okay, the user asked for "An animal, a symbol of strength in Brazil." Let me think about which animal that is. Brazil has several national symbols, but the most prominent one related to strength is the jaguar. Jaguars are apex predators, known for their power and agility. They're also the national animal of Brazil, right? Wait, actually, the national animal of Brazil is the jaguar, but sometimes people might confuse it with the cockatoo or other animals. Let me confirm. Yes, the jaguar is the national animal of Brazil and is often associated with strength and power. So the user probably wants an image of a jaguar. I should make sure to specify that it's a jaguar in the caption. The instruction should be clear: generate an image of a jaguar in its natural habitat, showcasing its muscular build and intense gaze to emphasize strength. Let me check if there's any other animal that could be considered. Maybe the capybara? No, capybaras are more known for being the largest rodents, not symbols of strength. So jaguar is definitely the right choice. **</think>** Generate an image of a majestic jaguar standing powerfully in the Brazilian rainforest, showcasing its muscular build, intense gaze, and distinctive spotted coat to symbolize strength and wilderness.	
Most commonly used tools during Diwali in India		**<think>** Okay, the user is asking about the most commonly used tools during Diwali in India. Let me start by recalling what Diwali is. It‘s the Festival of Lights, celebrated with lights, fireworks, and various traditions. But the user specifically mentioned “tools,” which is a bit tricky because Diwali isn’t typically associated with tools. Maybe they mean items or objects used in the celebrations. The main ones are diyas, rangoli materials (colored powders, rice, flowers), fireworks (firecrackers), sweets (like laddu, jalebi), and decorations (torans, lights). So, the response should clarify that while Diwali isn‘t about tools, the common items include diyas, rangoli materials, fireworks, sweets, and decorations. Then, if they meant cleaning tools, mention those as well. But the user said “most commonly used tools,” so maybe they’re referring to the items used in the rituals. **</think>** Generate an image showing a traditional Indian Diwali celebration scene with a woman lighting clay diyas (lamps) on a decorated home entrance, arranging colorful rangoli patterns with turmeric and rice powder, and preparing sweets in a kitchen with copper utensils, while fireworks and diyas illuminate the background.	
Eggs which are recently hatched		**<think>** Okay, the user asked for “Eggs which are recently hatched.” Let me break this down. First, they want images of eggs that have just hatched. But wait, eggs that have hatched would typically show the baby animal emerging, right? So maybe they‘re looking for images of newly hatched chicks, ducklings, or other baby animals coming out of their shells. I need to generate an image based on that. The correct approach would be to create an image showing a newly hatched chick (or similar) emerging from its eggshell. The caption should clarify that it’s a baby bird (like a chick) just hatched, with the eggshell partially broken. Alternatively, if it‘s a different animal, but the most common example is a chicken. The user didn’t specify the animal, so using a chick is safe. Let me confirm if there‘s any other possibility. Maybe ducklings or other birds, but chick is the standard example. **</think>** Generate a realistic image of a newborn chick with partially broken eggshell, standing on a nest of straw and twigs, showcasing soft downy feathers, bright yellow beak, and curious expression under natural daylight.	
A field of dandelions in autumn		**<think>** Okay, the user asked for “A field of dandelions in autumn.” Let me break this down. First, dandelions in autumn—dandelions are typically associated with spring, but they do bloom in autumn too, especially in temperate regions. The user probably wants a realistic scene showing dandelions during fall. I need to consider the key elements: a field, dandelions, autumn setting. Autumn means colors like gold, orange, brown, maybe some remaining green. Dandelions in autumn might have seed heads (the fluffy white balls) since they‘re late bloomers. The field should look natural, maybe with some grasses, fallen leaves, and maybe a few trees in the background to set the season. Let me check: dandelions bloom from spring to fall, so in autumn, they can have both flowers and seed heads. So the image should have both. Maybe some yellow flowers and some white seed heads. The field should look vibrant but with autumn colors. **</think>** Generate a realistic autumn landscape featuring a sunlit field of dandelions in various stages of bloom (yellow flowers and fluffy seed heads), surrounded by golden grasses, scattered fallen leaves, and soft sunlight highlighting the textures of the plants and earthy terrain.	
The instrument used by the Italian astronomer who famously observed the moons of Jupiter in the 17th century		<think> Okay, the user is asking about the instrument used by the Italian astronomer who observed Jupiter‘s moons in the 17th century. Let me recall who that astronomer is. It’s Galileo Galilei, right? He‘s famous for his telescopic observations of Jupiter’s moons, which were a big deal back then. Now, the instrument he used. I think it was a telescope, but I need to be specific. Galileo didn‘t invent the telescope, but he improved it and used it for astronomical observations. The key here is to mention the type of telescope he used. I remember he used a refracting telescope, which uses lenses to gather and focus light. I should phrase the response as: “Generate an image of a Galilean telescope, the refracting telescope used by Galileo Galilei for his observations of Jupiter’s moons in the 17th century.” That‘s precise and includes the key details. </think> Generate an image of a Galilean telescope, the refracting instrument used by Galileo Galilei for his groundbreaking 17th-century observations of Jupiter's four largest moons.	

Figure 4. Comparison of image generation with thinking versus without.

Input Image	Instruction	Results	Results *w*/Thinking	
	Draw what this species look like more than two million years ago		**<think>** Got it, the user is asking to draw what this species (a modern human) looked like more than two million years ago. The user said "this species" referring to the person in the image, which is a modern human. So the task is to imagine what the ancestor of modern humans looked like 2 million years ago. So the correct species is Homo erectus (since Homo sapiens didn't exist then). So the edit instruction should be: "Replace the modern human's features with those of Homo erectus, including a more robust build, prominent brow ridges, a larger nose, a smaller braincase relative to body size, and a more primitive facial structure, while maintaining the same pose and background."**</think>** Edit the image to depict a Homo erectus ancestor from approximately 2 million years ago, replacing the modern human's features with characteristics like a more robust skeletal structure, prominent brow ridges, a larger nasal aperture, a smaller braincase relative to body size, and a more primitive facial structure while retaining the same pose and background.	
	Draw what it will look like after two years underwater		**<think>** Got it, the user wants to know what the key would look like after two years underwater. First, I need to consider the effects of water on metal. The key would have significant corrosion: the metal would look dull, maybe with greenish or brownish patina (from copper corrosion), rust spots, and the edges might be eroded. The numbers "2600" might be partially worn. The key's surface would be rougher, with possible holes from corrosion. The wooden background might have water damage, but the key is the main subject. **</think>** Edit the image to show the key with significant corrosion, including a dull, tarnished metallic surface with greenish-brown patina, pitted edges, and partially eroded engravings (e.g., "2600" and the brand name), while the wooden background shows water damage and discoloration from prolonged submersion.	
	Draw what it will look like after one month of being neglected.		**<think>** Got it, the user wants to know what the plant will look like after one month of neglect. First, I need to analyze the current image. The plant is a potted houseplant, probably a type like a Pothos or maybe a similar broad-leafed indoor plant. Neglecting a plant for a month typically means no water, maybe no light, or both. Since it's on a windowsill, maybe the neglect is due to not watering. The plant in the image has glossy leaves, so it's likely a plant that needs regular watering. After one month without water, the soil would be dry, roots might start to die, leading to wilting. The leaves could lose their sheen, become limp, and some might turn yellow or brown. Maybe the plant starts to droop, and the overall appearance becomes less vibrant. **</think>** Edit the image to show the plant with wilted, yellowing leaves, brown edges, drooping stems, dry soil, and a slightly dusty pot, indicating one month of neglect.	
	Draw what it will look like after the knot is untied.		**<think>** Got it, the user wants to know what the image would look like after the knot is untied. First, I need to analyze the current image. It's a close-up of a complex, tightly knotted rope, probably a type of knot like a bowline or a similar secure knot. The rope is light brown, braided, and the knot is intricate with multiple loops and twists. When you untie a knot, the rope would return to its original, unknotted state. So the image after untying would show the rope as a straight, continuous braid without any knots. The background is plain white, so the unknotted rope would just be the same rope laid out straight, maybe with some slight natural curves from how it was handled, but no complex knots. So the edit instruction should be: "Remove the complex knot from the rope, leaving the rope as a single, straight, braided strand with natural fibers, against the same plain white background." **</think>** Remove the intricate knot from the rope, leaving the rope as a single, straight, braided strand with natural fibers and slight texture, against the same plain white background.	
	Draw what it will look like immediately after someone stands up from sitting on it for a long time.		**<think>** Got it, the user wants to know what the bean bag looks like after someone stands up from sitting on it for a long time. First, I need to visualize the current state. The bean bag is a lime green, plush, and has some creases from sitting. When someone sits on it, the fabric compresses, especially in the seat area. After standing up, the fabric would sag a bit more because the weight was removed, but the creases might remain. The bean bag might have a more pronounced dip or indentation where the person was sitting, and the overall shape might be less rounded, with some wrinkles or folds left behind. Also, the fabric might not fully return to its original shape immediately, so there could be temporary sagging. **</think>** Edit the image to show the lime green bean bag with visible sagging and deeper creases in the seat area, where the fabric has settled into a more pronounced indentation and relaxed folds after the person stood up, while maintaining the overall soft, plush texture and subtle wrinkles from prolonged sitting.	
	Draw what it will look like after popping.		**<think>** Got it, the user provided an image of a corn kernel and asked to draw what it will look like after popping. First, I need to recall that a popped corn kernel becomes a fluffy popcorn. The original image is a single yellow corn kernel. After popping, it should transform into a light, airy, and expanded structure with a white or pale yellow color, with a soft, porous texture. The shape would be irregular and puffy, not the hard, smooth oval of the raw kernel. I need to describe the edit instruction clearly. Let me make sure to capture the key elements: the transformation from a small, hard kernel to a large, fluffy, white/pale yellow popped piece with a soft, airy appearance. **</think>** Edit the image to transform the single yellow corn kernel into a fluffy, expanded popcorn kernel with a light golden-brown, airy texture, showing the characteristic puffed, irregular shape and soft, porous surface typical of popped corn.	
	Draw what it will look like after being hit by a baseball.		**<think>** Got it, the user wants me to describe what the window would look like after being hit by a baseball. First, I need to visualize the original image: a window with wooden frames, curtains, plants inside. A baseball hit would cause damage. Let's think about the impact. The window is probably glass, so the baseball would shatter the glass. The wooden frame might get dented or cracked, but the main damage is to the glass. The window sill might have debris from the broken glass. I need to make sure the description is clear and accurate. Let me structure it: the glass is shattered with cracks radiating from the impact point, pieces of glass scattered, curtains torn, plants disturbed, frame possibly cracked. **</think>** Shatter the central glass pane of the window with radial cracks, scatter small glass fragments across the sill and frame, tear the curtains slightly, and displace the indoor plants while keeping the wooden frame intact but showing visible impact marks.	

Figure 5. Comparison of image editing with thinking versus without.

References

[1] David Beniaguev. Synthetic faces high quality - text 2 image (sfhq-t2i) dataset, 2024. 1

[2] Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. Sharegpt-4o-image: Aligning multimodal models with gpt-4o-level image generation. *arXiv:2506.18095*, 2025. 3

[3] Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. Sharegpt-4o-image: Aligning multimodal models with gpt-4o-level image generation, 2025. 1

[4] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv:2505.09568*, 2025. 3

[5] Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, et al. Paddleocr 3.0 technical report. *arXiv preprint arXiv:2507.05595*, 2025. 2

[6] Nikai Du, Zhennan Chen, Shan Gao, Zhizhou Chen, Xi Chen, Zhengkai Jiang, Jian Yang, and Ying Tai. Textcrafter: Accurately rendering multiple texts in complex visual scenes. *arXiv preprint arXiv:2503.23461*, 2025. 2

[7] Rongyao Fang, Aldrich Yu, Chengqi Duan, Linjiang Huang, Shuai Bai, Yuxuan Cai, Kun Wang, Si Liu, Xihui Liu, and Hongsheng Li. Flux-reason-6m & prism-bench: A million-scale text-to-image reasoning dataset and comprehensive benchmark. *arXiv preprint arXiv:2509.09680*, 2025. 1

[8] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36:27092–27112, 2023. 1

[9] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023. 2

[10] Maksim Kuprashevich, Grigorii Alekseenko, Irina Tolstykh, Georgii Fedorov, Bulat Suleimanov, Vladimir Dokholyan, and Aleksandr Gordeev. NoHumansRequired: Autonomous High-Quality Image Editing Triplet Mining. *arXiv preprint arXiv:2507.14119*, 2025. 1

[11] Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, et al. Uniworld: High-resolution semantic encoders for unified visual understanding and generation. *arXiv:2506.03147*, 2025. 2, 3

[12] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025. 1

[13] Jian Ma, Xujie Zhu, Zihao Pan, Qirong Peng, Xu Guo, Chen Chen, and Haonan Lu. X2edit: Revisiting arbitrary-instruction image editing through self-constructed data and task-aware representation learning, 2025. 1

[14] Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15086–15095, 2025. 2

[15] Yuwei Niu, Munan Ning, Mengren Zheng, Weiyang Jin, Bin Lin, Peng Jin, Jiaqi Liao, Chaoran Feng, Kunpeng Ning, Bin Zhu, et al. Wise: A world knowledge-informed semantic evaluation for text-to-image generation. *arXiv preprint arXiv:2503.07265*, 2025. 1

[16] OpenAI. Introducing 4o image generation, 2025. 1

[17] Yusu Qian, Eli Bocek-Rivele, Liangchen Song, Jialing Tong, Yinfei Yang, Jiasen Lu, Wenze Hu, and Zhe Gan. Pico-banana-400k: A large-scale dataset for text-guided image editing, 2025. 1, 2

[18] Yiren Song, Cheng Liu, and Mike Zheng Shou. Omniconsistency: Learning style-agnostic consistency from paired stylization data. *arXiv preprint arXiv:2505.18445*, 2025. 1

[19] Qwen Team. Qwen3-vl, https://github.com/qwenlm/qwen3-vl. 2025. 1

[20] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 1, 2

[21] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. 2019. 1

[22] Yuhan Wang, Siwei Yang, Bingchen Zhao, Letian Zhang, Qing Liu, Yuyin Zhou, and Cihang Xie. Gpt-image-edit-1.5 m: A million-scale, gpt-generated image dataset. *arXiv preprint arXiv:2507.21033*, 2025. 1

[23] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv:2506.18871*, 2025. 1

[24] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 1

[25] Junyan Ye, Dongzhi Jiang, Zihao Wang, Leqi Zhu, Zhenghao Hu, Zilong Huang, Jun He, Zhiyuan Yan, Jinghua Yu, Hongsheng Li, et al. Echo-4o: Harnessing the power of gpt-4o synthetic images for improved image generation. *arXiv preprint arXiv:2508.09987*, 2025. 1

[26] Haozhe Zhao, Xiaojian Shawn Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. In *NeurIPS*, 2024. 1

[27] Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15329–15339, 2025. 1