Dual-Process Image Generation

Supplementary Material

7. Ablations for Commonsense Inferences

In this section, we characterize the discriminative ability of different VLMs, and its effect on our method's generation quality.

VLM Scoring. First, we compare the agreement between VLM scores and human annotations. We take 1200 images generated by Flux Schnell, which were released by CommonsenseT2I[1]. We have one human annotator then rate each image as correct/incorrect. We also feed these same images to different open-source VLMs, prompted with: *Does the image generally fit the description "`<description>`"? Answer with Yes or No.* We then compare the alignment of the human and VLM ratings, using the metrics of accuracy and sensitivity. To compute the classification accuracy, we compare the most likely next token with the human annotated ground-truth. For sensitivity, we extract the probabilities of the "Yes" and "No" tokens from the vocabulary distribution, and compute the difference between the correct vs. incorrect class (as labeled by the human annotations). We report the results in Table 3, for both newer and bigger open-source VLMs, as well as the closed-source GPT4o upper bound. Evidently, newer VLMs like Idefics2 and bigger VLMs like Gemma3 27B strongly agree with human answers, with less than a 1% difference from GPT4o. We also see that our method performs better with stronger VLMs; supervising with Idefics2 instead of LLaVA leads to a 14.5% boost on CommonsenseT2I generations.

	newer models →			bigger models →			
	LLaVA [46]	Qwen2.5VL [4]	Idefics2 [40]	Gemma3 4B [74]	Gemma3 12B	Gemma3 27B	GPT4o [56]
Accuracy (↑)	0.7250	0.7892	**0.8092**	0.7758	0.7858	**0.8108**	0.8183
Sensitivity (↑)	0.2875	0.5504	**0.5998**	0.5536	0.5686	**0.6219**	N/A

Table 3. **Newer and bigger VLMs agree more with humans on image correctness.** We report both the accuracy and sensitivity of different VLM scores compared with human annotations on CommonsenseT2I.

Automatic Question Reliability. Next, we ablate the influence of the VLM question on the generator output. In our normal automatic question generation pipeline, we feed only the prompt to GPT4o, which correctly covers the desired commonsense inference 82% of the time. Here, we also experiment with more "reliable" questions derived from the ground-truth inferences provided by CommonsenseT2I, by converting these inferences into questions with GPT4o. In Table 4 we observe that simply improving the supervision signal with more reliable questions, without modifying the fine-tuning procedure, can result in progressively larger gains in generation quality.

	percentage of reliable questions →				
	0%	25%	50%	75%	100%
Accuracy %Δ from No Control (↑)	+20.1	+23.9	+24.8	+27.1	**+28.2**

Table 4. **More reliable VLM questions lead to better generator outputs.** We ablate guiding varying proportions of samples with more reliable questions derived from the ground-truth labels from CommonsenseT2I, mixed with automatically generated questions. We report results in terms of the performance delta on top of the base prompt, or "No Control."

Image Quality Comparison. Finally, we validate that our method's improvements in logical accuracy do not compromise image quality in Figure 11. Following the setup from Jayasumana et al. [31], we use MS-COCO 30K [43] as the real image set and outputs from CommonsenseT2I [22] as the generated image set. Evidently, our method matches – and in fact exceeds – the image quality of the prompting baselines across the board. This improvement could be because low perceptual quality also hurts logical coherence; for example, if a water glass is malformed then it is also unlikely to be recognized as correctly "spilling [...] onto the table."

[1]The images can be found at https://zeyofu.github.io/CommonsenseT2I/visualization_flux_schenel.html.

Generator	Method	CMMD (↓)	FID (↓)
Flux Schnell (single-step)	Base Prompt	1.17	102.8
	Base Prompt + Ours	**1.10**	**98.7**
	Expanded Prompt	1.23	102.0
	Expanded Prompt + Ours	**1.15**	**98.7**
Flux Dev (multi-step)	Base Prompt	1.26	99.6
	Base Prompt + Ours	**1.21**	**97.7**
	Expanded Prompt	1.27	99.5
	Expanded Prompt + Ours	**1.23**	**97.6**

Figure 11. **Image Quality Comparison.** For the same samples evaluated in Table 1, we also compute their image quality, measured via CLIP Maximum Mean Discrepancy (CMMD) [31] and Frechet Inception Distance (FID) [27, 59]. We compute the FID between the 1.2k generated images and 30k real images from MS-COCO [43].

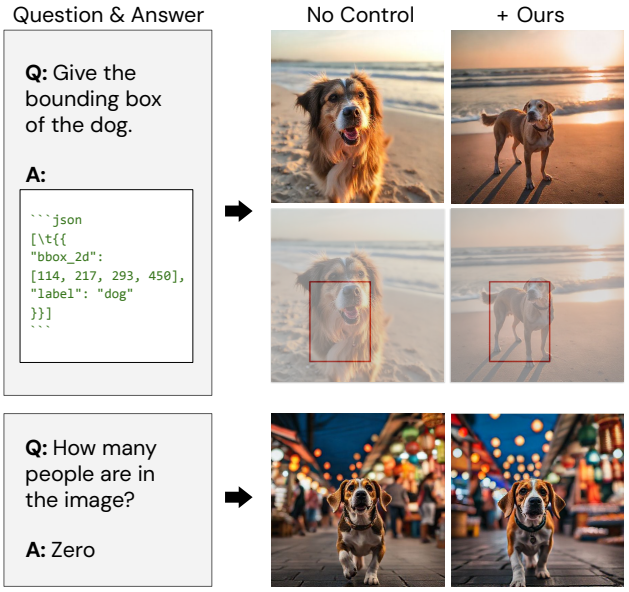

Figure 12. **Beyond Yes-No Questions.** Visual controls can also be implemented by setting the answer to the desired coordinates (for bounding box control) or count (for object deletion). Both examples are shown on Qwen2.5-VL; we use its default bounding box template format.

8. Additional Results for Visual Prompting

Other Visual Controls. In Figure 12 we show additional visual controls that can be implemented with our method. See Table 6 for the image generator prompts. Our method is not limited to Yes-No questions; the same VQA loss can also optimize other answer strings like bounding box coordinates or object counts. For these examples, we use Qwen2.5-VL, which is specifically trained for bounding box prediction and counting. Other VLMs like Idefics2 struggle with precise object grounding.

Horizon Control Evaluation. Here, we design an automatic evaluation for horizon control. We fine-tune LoRAs for each horizon line, and estimate the horizon of the generated images. For our inputs, we take the cross product of three prompts from Figure 6 and five lines approximately equally spaced across the image height. For each LoRA, we generate images with five unseen seeds. We use Perspective Fields [33] to compute per-pixel latitude and longitude maps, then extract the horizon from the zero-latitude line. We compute the "horizon distance" as the difference between the estimated and control horizons, normalized by the image height. We also validate the quality of this automatic metric by manually annotating 100 images, where we find the mean error is 0.0470. We report the results in Table 5, where we find that our method outperforms prompt expansion by 0.0218 points.

	No Control	Prompt Expansion	Ours
Horizon Distance (↓)	0.2204	0.1923	**0.1705**

Table 5. **Quantifying horizon control.** On 75 images (3 prompts×5 controls×5 seeds), we report the horizon distance (bounded between 0 and 1) between the generated image and control.

Additional Results. In Figures 15-19 we display extended results from our learned weights, run on five random seeds.

Image Generator Prompts. In Table 6 we display the prompts associated with each figure in Sec. 4.3 of the main text.

Task	Figure	Prompts
Color Palette	Figure 4	*"Pop art style household objects"*, *"A pixel art sunset"*, *"A colorful living room."*
Line Weight	Figure 5	*"Cartoon of a cat against a plain background."*
Horizon Position	Figure 6	*"A teddy bear riding a skateboard in Times Square"*, *"A dutch landscape painting"*, *"A tropical beach"*
Relative Depth	Figure 7	*"A dog and a sheep."*
Visual Composition	Figure 8	*"A painting of an oceanscape with a lighthouse, cruise, and seal colony"*, *"A painting of a Dutch landscape with a tree, windmill, and wheat field"*
Bounding Box & Object Removal	Figure 12	*"A dog at the beach"*, *"A beagle walking through a lively night market"*

9. Templates and System Prompts for VLMs

Template Formatting. As discussed in Sec. 3, input formatting is important for obtaining meaningful scores from the VLM. In Figures 13, 14 we display example inputs for different open-source VLMs, following their officially recommended templates.

User: <image> Is the butter melting in the pan? Answer with Yes or No.<end_of_utterance>
Assistant: Yes

Figure 13. An example (image, question, answer) triplet formatted for Idefics2 [40].

<|im_start|>**system**
You are a helpful assistant.
<|im_end|>
<|im_start|>**user**
<|vision_start|><|image_pad|><|vision_end|>
Is the egg yolk and white still in liquid form? Answer with Yes or No.
<|im_end|>
<|im_start|>**assistant**
Yes

Figure 14. An example (image, question, answer) triplet formatted for Qwen2.5-VL [4].

Prompt Expansion and Automatic Questions. Here, we include additional details related to our commonsense understanding evaluation Sec. 4.2. For the prompt expansion baseline, we use the same system prompt used for DALLE-3 [6], from Liu [47]. For the automatically generated questions used by our method, we design a system prompt shown in Figure 20.

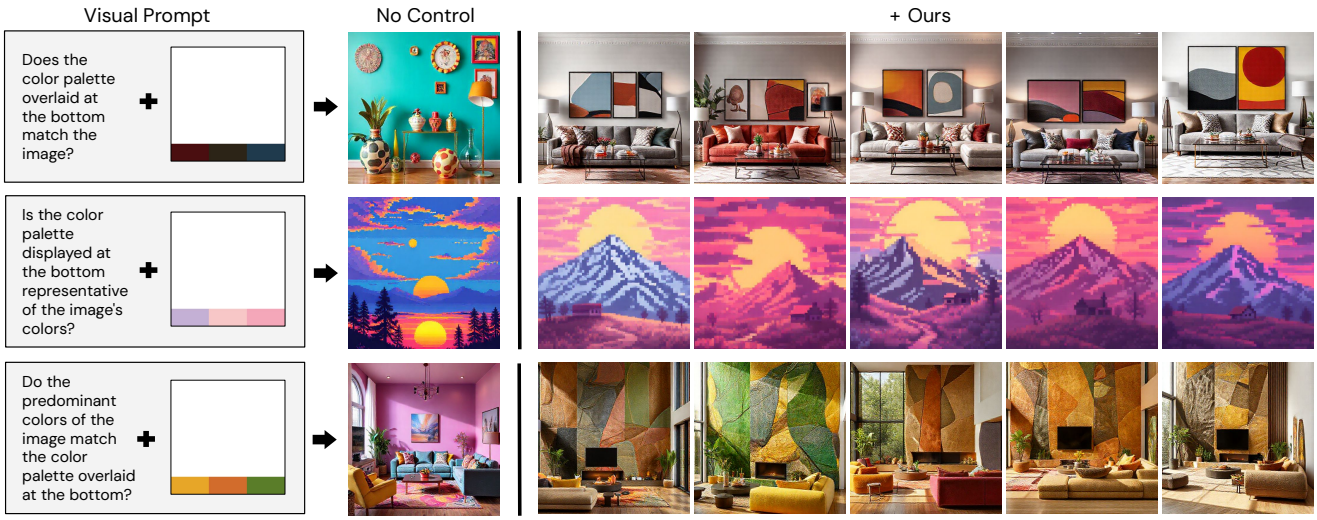

Figure 15. **Color Palette.** Corresponding to Figure 4 in the main text, we show expanded results for a single set of weights optimized with our method, run on five random seeds.

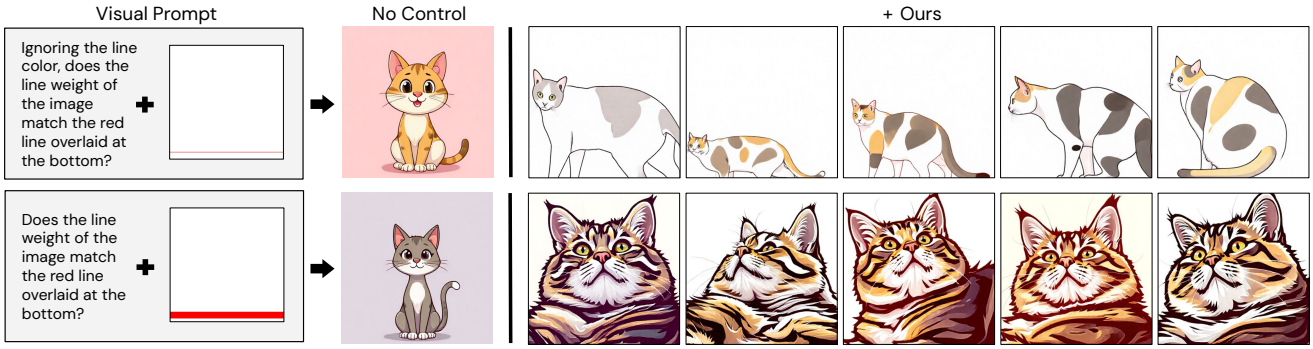

Figure 16. **Line Weight.** Corresponding to Figure 5 in the main text, we show expanded results for a single set of weights optimized with our method, run on five random seeds.

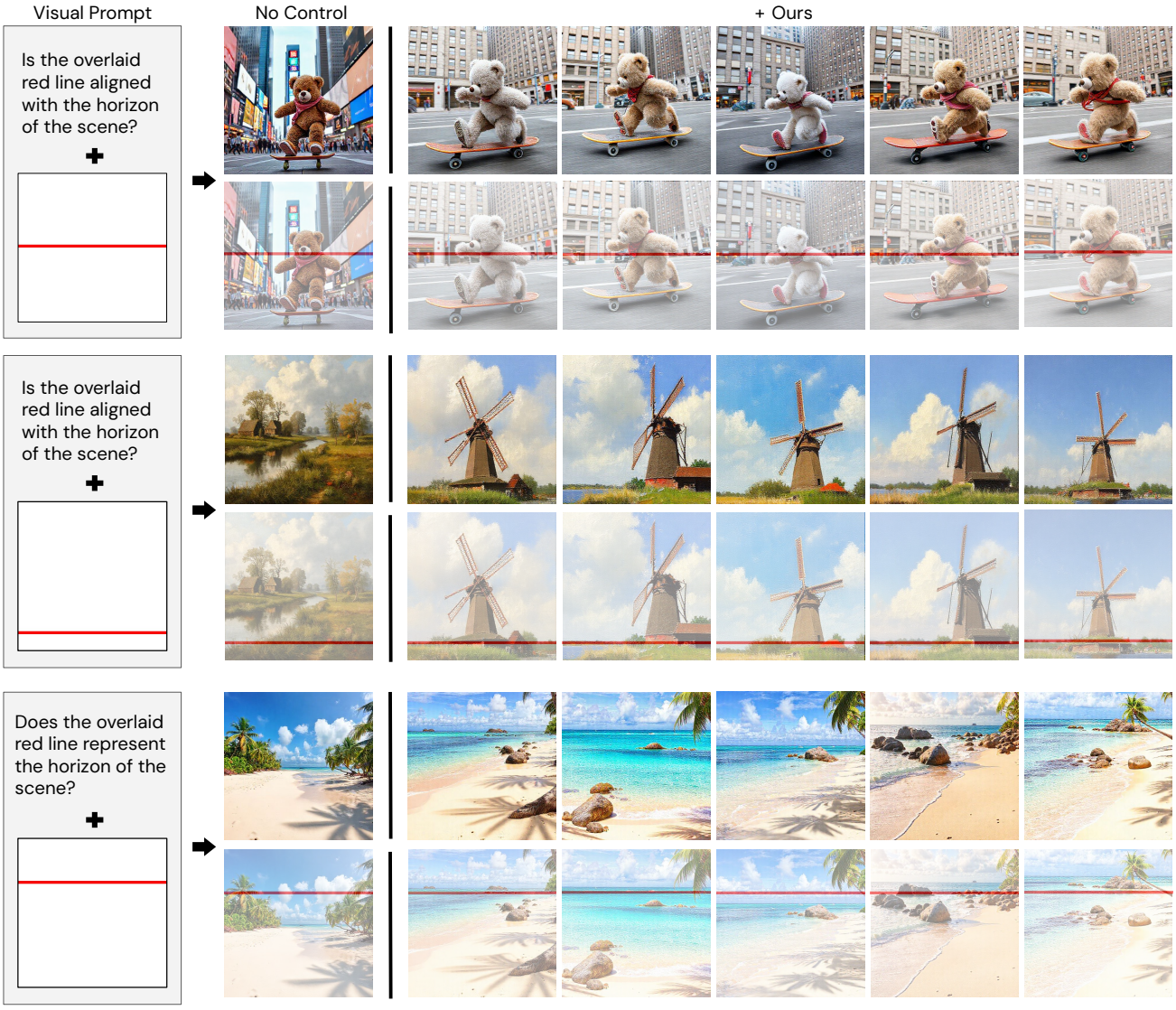

Figure 17. **Horizon Position.** Corresponding to Figure 6 in the main text, we show expanded results for a single set of weights optimized with our method, run on five random seeds.

Figure 18. **Relative Depth.** Corresponding to Figure 7 in the main text, we show expanded results for a single set of weights optimized with our method, run on five random seeds.

Figure 19. **Visual Composition.** Corresponding to Figure 8 in the main text, we show expanded results for a single set of weights optimized with our method, run on five random seeds.

You are an expert image verification assistant. Your task is to write questions that evaluate whether a generated image accurately reflects a given text prompt.

Instructions

1. You will be provided a prompt, and you will generate a list of questions.
2. Each prompt is underspecified, in order to test a commonsense inference.
3. Think about the most likely inference the prompt is trying to test. This inference is related to physical laws, human practices, biological laws, daily items, and animal behaviors.
4. Write a specific label for this inference.
5. Write a question that checks this inference.
6. The expected answer for all questions should be "Yes".
7. Generate up to N=1 label-question-answer triplets. Express your output in JSON format as a list of question-answer pairs `[[label1, question1, answer1], [label2, question2, answer2], ...]`

Examples

Labels and questions should be simple.
They should mention a minimal number of visible objects.
They should also pay attention to wording, for example, if an action is about to happen or has already happened.

Prompt: A straw submerged in a glass of water
Output:

```
["Objects appear bent due to refraction.",
 "Does the straw look bent underwater?",
 "Yes"]
```

Prompt: A firefighter on holiday
Output:

```
["People wear casual clothes when they are off-duty.",
 "Is the person wearing casual clothes?",
 "Yes"]
```

Prompt: An orange tree in the springtime
Output:

```
["Orange trees have white blossoms when blooming.",
 "Does the tree have white blossoms?",
 "Yes"]
```

Prompt: A balloon filled with air, tied to a fence
Output:

```
["The balloon sinks because its air is the same density as the outside air.",
 "Is the balloon on the ground?",
 "Yes"]
```

Prompt: A bird taking a nap
Output:

```
["Birds sleep perched on branches in their natural habitat.",
 "Is the bird asleep on a branch?",
 "Yes"]
```

Your Turn

Prompt: `<prompt>`
Output:

Figure 20. Here we display our system prompt used to automatically generate questions, to check commonsense inferences. To produce higher quality inferences, we include manually created in-context examples of Prompt-Output pairs separate from, but inspired by, the CommonsenseT2I [22] benchmark.