6. Overview of Supplementary

The supplementary material is organized into the following sections:

- Section 7: Implementation details.
- Section 8: More experiments and analysis.
- Section 9: More analysis on diffusion generation prior.
- Section 10: Detailed prompt for generation prior.
- Section 12: Discussion and limitation.
- Section 11: Human evaluation guideline.
- Section 13: More visualization comparison and results.

7. Implementation Details

We implemented our editing model training based on the InstructPix2Pix PyTorch [35] code from the Diffusers repository [48], using Stable Diffusion V1.5 [42] as the pre-trained weights for the editing model. Following InstructPix2Pix's implementation [4], we enable classifier-free diffusion guidance [20] for both the image condition and the text condition with 5% mask probability during training. The training batch size is 512 with a learning rate of 1e-4, weight decay of 1e-2, and a warm-up ratio of 100 steps. The training resolution is 512x512 by resizing input images without any crops. Margin $m = 5e-3$ and weight $\lambda = 1.0$ is used for triplet loss $\mathcal{L}_{\text{triplet}}$. We train the edit model for 10,000 steps and use the triplet loss after the 2,000 training steps. During inference, we keep the original image ratio and resize the shorter side to 512, with DDIM [49] sampler and 50 sampling steps, following the default settings of Multi-Reward [16]. The text guidance scale and image guidance scale we used for inference are 10.0 and 1.5, respectively.

8. More Experiments and Analysis

In this section, we provide more experiments and analysis. We first present the MagicBrush benchmark results in Sec. 8.1, and more benchmark results in Sec. 8.2, and finally analyze GPT-4o cost and different VLMs in Sec. 8.3.

8.1. Evaluation on MagicBrush Benchmark

In Tab. 5, we present a quantitative comparison of various image editing methods evaluated on the MagicBrush single-turn benchmark. However, it's important to note that these automated metrics (CLIP-I, CLIP-T, DINO, L1) should be interpreted with caution. As highlighted by previous works [16, 24, 45], such metrics often fail to fully capture human perceptual preferences, and can sometimes lead to misleading conclusions about actual editing quality. Several studies have demonstrated significant discrepancies between metric-based rankings and human evaluation results [16, 24, 45].

Our proposed method adopts a data-oriented approach, contrasting with the model-oriented strategies prevalent in image editing. Remarkably, without requiring additional parameters, pretraining tasks, or extensive training data (using only 40K samples compared to 300K-1.2M in other methods), our approach achieves competitive performance across all metrics. The CLIP-T score of 30.3 is only 0.3 lower than the best results, and DINO score of 80.2 (second highest) is particularly noteworthy, suggesting strong preservation of both semantic and structural image features.

Method	Extra Module	Pretrain Tasks	Edit Data	CLIP-I↑	CLIP-T↑	DINO↑	L1↓
InstructPix2Pix [4]	✗	✗	300K	85.4	29.2	69.8	0.112
InstructDiffusion [14]	✗	✓	860K	89.2	30.2	77.7	-
MagicBrush [58]	✗	✗	310K	**90.7**	**30.6**	**80.6**	**0.062**
SmartEdit [24]	✓	✓	1.2M	90.4	30.3	79.7	<u>0.081</u>
SuperEdit (Ours)	✗	✗	40K	<u>90.5</u>	<u>30.3</u>	<u>80.2</u>	0.106

Table 5. Quantitative comparison (L1/CLIP-I/CLIP-T/DINO-I) on the MagicBrush benchmark. Our SuperEdit achieves good performance with better efficiency, without extra modules or pretrain tasks.

8.2. Other Benchmarks

As shown in Tab. 6, we present a comparative evaluation of different editing methods on the Real-Edit benchmark, using Gemini-1.5 Pro as the judge. The assessment is conducted across three core dimensions: Following, which measures the fidelity of the edited image to the text instruction; Preserving, which evaluates the preservation of non-edited regions; and Quality, which assesses the overall visual appeal of the resulting image. The results unequivocally demonstrate that the SuperEdit method significantly outperforms both InstructP2P and SmartEdit across all evaluated metrics. Specifically, SuperEdit achieves the highest scores for instruction following (75%, 3.97), content preservation (80%, 4.26), and overall quality (71%, 4.13), underscoring its superior editing capabilities and reliability.

Method	Following↑	Preserving↑	Quality↑
InstructP2P	61%, 3.29	60%, 3.46	57%, 3.60
SmartEdit	70%, 3.85	65%, 3.81	61%, 3.62
SuperEdit	**75%, 3.97**	**80%, 4.26**	**71%, 4.13**

Table 6. Gemini-1.5 Pro evaluation results on the Real-Edit benchmark.

We also present the quantitative comparison of our method against the IP2P and SmartEdit baselines on the EmuEdit benchmark in Tab. 7. A key highlight is the remarkable efficiency of our approach: our method achieves these results using only 40K training pairs and a 1.1B parameter model, making it significantly more lightweight in terms of both data requirements and model size. Despite its lightweight nature, our method achieves state-of-the-art performance on four out of the five key metrics. Specifically, it achieves leading scores in text-image alignment (CLIP dir: 0.103), background preservation (CLIP$_{img}$: 0.848), output quality (CLIP$_{out}$: 0.235), and feature similarity (DINO: 0.800). These results strongly validate the superior performance and high efficiency of our proposed method on the EmuEdit benchmark.

8.3. GPT-4o Cost & Different VLMs' Performance

We respectfully emphasize that our core contribution is identifying and addressing noisy supervision in existing datasets, rather than focusing on cost-effective scaling strategies. Using

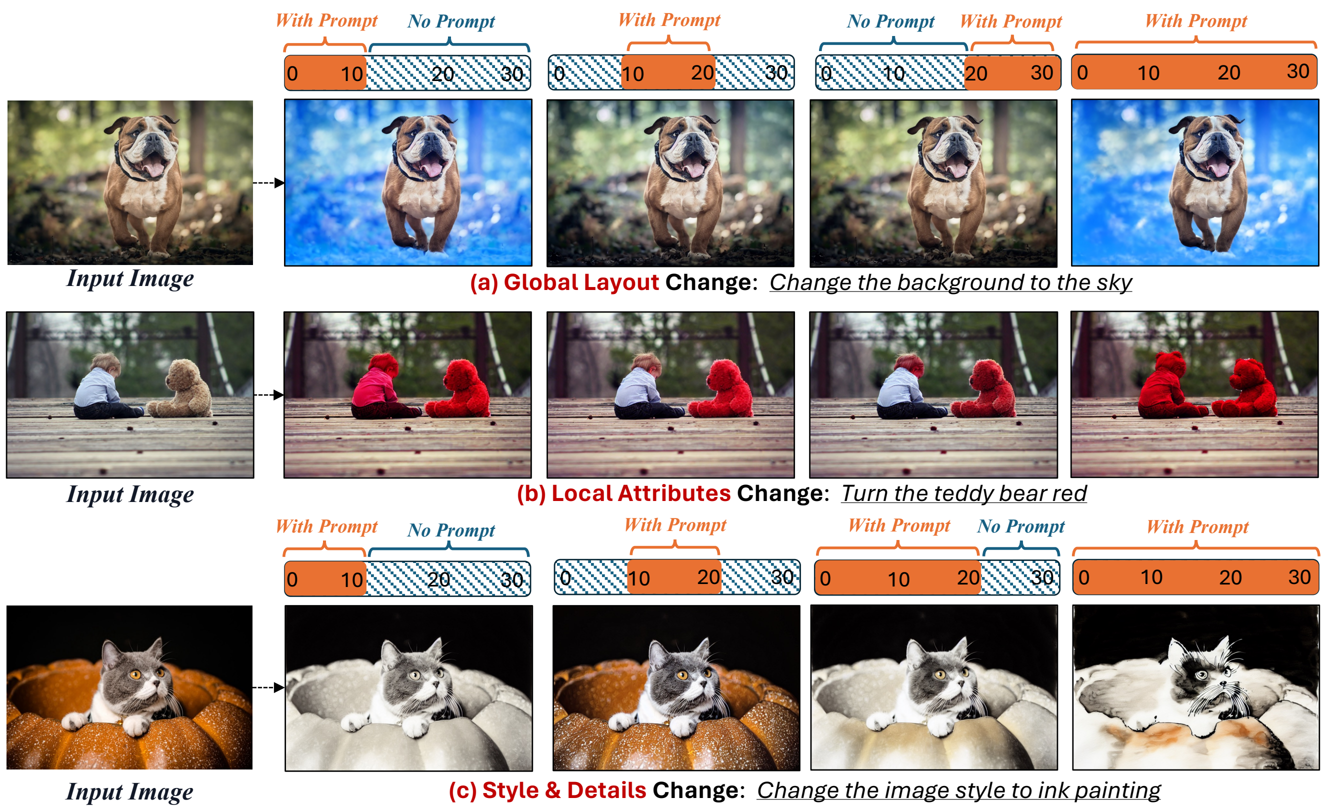

Figure 8. We show the impact of incorporating the editing prompt at different inference timesteps on the edited image. **(a)** The global layout changes usually occur in the early stages of inference. Adding text editing instructions to modify the global layout at the mid or late stages does not effectively impact the global layout. **(b)** Local object attribute changes occur in the mid-stages of sampling. Adding text editing instructions in the early or late stages may result in incorrect editing outcomes. **(c)** The style changes happen across all inference stages, and the detail changes happen in the late stage (Please refer to the subtle differences between the last two images). Best viewed in color.

Method	Edit Data	Model Size	EmuEdit Bench				
			$\text{CLIP}_{\text{dir}}\uparrow$	$\text{CLIP}_{\text{im}}\uparrow$	$\text{CLIP}_{\text{out}}\uparrow$	L1↓	DINO↑
IP2P	400K	1.1B	0.078	0.834	0.219	0.121	0.762
SmartEdit	1.2M	14.1B	0.101	0.838	0.231	**0.101**	0.792
Ours	**40K**	**1.1B**	**0.103**	**0.848**	**0.235**	0.112	**0.800**

Table 7. Comparison on the EmuEdit benchmark.

Model	Following↑	Preserving↑	Quality↑
LLaVA-OV(72B)	37%, 2.23	48%, 2.68	39%, 2.40
Qwen2-VL(72B)	33%, 2.11	45%, 2.60	35%, 2.22
GPT-4o	**49%, 2.87**	**60%, 3.71**	**55%, 3.69**

Table 8. Real-Edit results with 5k randomly sampled training data.

GPT-4o	LLaVA-OV(72B)	InternVL2(76B)	Qwen2-VL(72B)
76.2%	50.4%	48.2%	47.8%

Table 9. Instruction rectification success rate across 100 samples

GPT-4o for our method costs \$0.02 per 512×512 input-edited image pair, totaling \$800 for 40K data, which is less expensive than existing works that require additional VLM fine-tuning or extra pre-training stages. As GPT-4o is the best solution (Tab.6 in the supplementary), we use it to validate our methods and open-source these valuable data to the community. We also provide results on improving editing instructions with open-source models in Tab.8. In addition, we asked 5 annotators to evaluate rectified instructions from different VLMs. As shown in Tab. 9, existing open-source VLMs can partially substitute GPT-4o. These open-source models can be further fine-tuned with GPT-4o data and then used for efficient scaling up, which we leave for future work.

9. Diffusion Generation Prior

As discussed in Sec. 3.2 and Fig. 4 of the main paper, editing diffusion models focus on specific generation attributes during inference, independent of the different editing instructions. Specifically, editing models focus on global layout in the early stages, local object attributes in the mid stages, image details in the late stages, and style change across all sampling stages. In this section, we further demonstrate this generation prior in Fig. 8.

Fig. 8 provides compelling visual evidence for the claims made in the main paper regarding how diffusion models process different aspects of image generation at specific timesteps. The experiments systematically demonstrate that this behavior is consistent across various editing tasks, reinforcing the observation that “different timesteps play distinct roles in image generation for text-to-image diffusion models, regardless of the text prompt” as cited in previous works.

Specifically, the figure illustrates three key patterns: (a) Global Layout Changes: The first row shows that changing the background to sky is most effective when prompts are introduced in the early stages (0-10 timesteps). When the same editing instruction is applied during mid (10-20) or late (20-30) stages, the model fails to properly modify the global layout, maintaining the original forest background despite the editing instructions. This validates our assertion that “diffusion models focus on global layout in the early stages.” (b) Local Object Attributes: The second row demonstrates that local attribute modifications, such as changing the teddy bear’s color to red, are optimally achieved during the mid-stages of sampling (10-20 timesteps). When the color change instruction is introduced too early or too late, the results show inconsistent or incomplete color transformation. This confirms that “local object attributes are processed in the mid stages”. (c) Style and Details: The third row reveals two important insights. First, style transformations (changing to ink painting style) can be effectively applied across all timesteps, indicating that style modifications have a more flexible temporal window. Second, subtle detail refinements are predominantly processed in the late stages (20-30), as evidenced by the finer differences between the last two images in the bottom row. This supports our claim about “image details in the late stages of sampling.” These observations not only validate the theoretical framework presented in the main text but also provide practical insights for optimizing instruction-based image editing. The clear temporal division of editing capabilities suggests that a more nuanced approach to prompt timing could significantly improve editing outcomes. This understanding directly supports our approach of guiding Vision-Language Models based on these four generation attributes (global layout, local attributes, style, and details), enabling us to establish a unified rectification method applicable across various editing instructions as described in the main paper.

10. GPT-4o Prompts for Constructing Rectified and Contrastive Editing Instructions

We show the detailed prompt for GPT-4o to construct the rectified and contrastive editing instructions in Fig. 9. As discussed in Sec. 9, we input the original image and the edited image into GPT-4o and ask it to return the differences in the following four attributes: “Overall Image Layout “Local Object Attributes”, “Image Details”, and “Style Change”. When calling the GPT-4o API, we explicitly define “Overall Image Layout” as modifications to the major objects, characters, and background in the image. “Local Object Attributes” are defined as changes in the texture, motion, pose, and shape of the major objects, characters, and background. Additionally, we combine “Style” and “Details” into a single category to reduce the number of tokens generated by GPT-4o, thus saving costs. We observed that this adjustment does not reduce GPT-4o’s understanding of the style and detail changes between the original-edited image pair. In the actual training of the editing model, acknowledging that CLIP [38] text encoder can accept a maximum of 77 textual tokens as input, we ask GPT-4o to summarize and refine these rectified instructions. We then use the consolidated and refined editing instructions (“Summarized Instruction” in Fig. 9) to train the model.

11. Human Evaluation Scoring Guidelines

Following: This metric assesses how well the edited image adheres to the text instruction to modify the original image.

- **Score 5 (Excellent):** Strictly follows the instruction. All specified modifications are accurately reflected without any omissions or deviations.
- **Score 4 (Good):** Largely follows the instruction. Most modifications are accurately executed, but there might be minor inaccuracies or slight deviations in subtle details with minimal impact.
- **Score 3 (Acceptable):** Generally follows the instruction, but some modifications are inaccurate or incomplete. Some requested changes are not fulfilled.
- **Score 2 (Poor):** Partially follows the instruction. Multiple specified modifications are incorrectly executed.
- **Score 1 (Very Poor):** Largely fails to follow the instruction. The result is inconsistent with the prompt, and the modifications are severely incorrect.
- **Score 0 (Failure):** Completely fails to follow the instruction. The result is entirely inconsistent with the desired outcome.

Preserving: This metric evaluates whether the unedited parts of the image—such as the background, and subject identity—are preserved consistently with the original image.

- **Score 5 (Excellent):** Perfectly preserved. All unedited regions remain identical to the original image without any unnecessary alterations. The subject’s identity, background, and textures are flawlessly maintained.
- **Score 4 (Good):** Almost fully preserved. Nearly all unedited regions are maintained, with only negligible changes to very minor details that have minimal impact.
- **Score 3 (Acceptable):** Mostly preserved. The majority of unedited regions are unchanged, but some areas exhibit slight, unintended modifications or minor discrepancies.
- **Score 2 (Poor):** Partially preserved. Several unedited regions show noticeable changes, and important details from the original image are improperly altered.
- **Score 1 (Very Poor):** Poorly preserved. The majority of unedited regions are significantly altered, leading to the loss or destruction of original details and key features.
- **Score 0 (Failure):** Not preserved at all. The unedited regions are completely changed, failing to retain any of the original image’s details or features.

Quality: This metric assesses the perceptual quality of the edited image compared to the original, focusing on significant degradation, structural errors, or unnatural artifacts.

- **Score 5 (Excellent):** The edited image quality is consistent with the original. The image is clear, free of any noticeable degradation, artifacts, or unnatural phenomena.
- **Score 4 (Good):** The image quality is largely maintained, with only slight degradation or minor, inconspicuous artifacts that barely affect the overall impression.
- **Score 3 (Acceptable):** The image shows noticeable degradation and some unnatural artifacts, but the overall result is still acceptable.
- **Score 2 (Poor):** The image suffers from severe degradation, with unnatural artifacts in multiple regions that significantly impact the viewing experience.
- **Score 1 (Very Poor):** The image is plagued with severe artifacts and unnatural content in almost all regions, rendering it unusable.
- **Score 0 (Failure):** The image quality is extremely low and severely distorted, making it unrecognizable or unusable.

12. Discussion and Limitation

Discussion. It's important to emphasize that our data-oriented approach is not mutually exclusive with model-oriented methods like MultiReward or SmartEdit, nor is its purpose to surpass existing work across various benchmarks or diminish their excellent contributions. Instead, our work explores a complementary yet important research question: What level of performance can be achieved with minimal architectural modifications by primarily focusing on supervision quality and optimization? Surprisingly, under both GPT-4o and human evaluation, our method significantly outperforms existing approaches despite using only a small amount of data, without modifying the model architecture, and requiring no additional pretraining. This suggests that high-quality data can substantially compensate for architectural simplicity, achieving results comparable to or even better than methods with considerably more parameters and pretraining requirements. We believe our approach and experimental results bring new insights and novelty to the field of image editing research.

Method	Pre-trained U-Net	Model Size Edit Data	Following ↑ Acc	Following ↑ Score	Preserving ↑ Acc	Preserving ↑ Score	Quality ↑ Acc	Quality ↑ Score
SmartEdit	InstrutDiff	14.1B/1.2M	64%	3.50	66%	3.70	45%	3.56
SuperEdit	SD1.5	1.1B/40K	67%	3.59	77%	4.14	65%	4.01
SuperEdit	InstrutDiff	1.1B/40K	**71%**	**3.76**	**83%**	**4.32**	**71%**	**4.17**

Table 10. SuperEdit outperforms the SOTA SmartEdit and achieves further improvements with InstructDiffusion pre-trained weights.

Furthermore, since our data-oriented approach is complementary and orthogonal to existing work, we can build upon current methods to further improve editing performance. Specifically, we follow the same setup as SmartEdit, retraining our model using InstructDiffusion as the pre-trained weights. The experimental results, as shown in Tab. 10, demonstrate that our method can complement existing work to achieve even better editing performance. When comparing SuperEdit with InstructDiffusion pre-trained weights against SmartEdit, we observe significant improvements across all metrics (71% vs. 64% in following instructions, 83% vs. 66% in preserving content, and 71% vs. 45% in image quality), despite using only 40K training samples compared to SmartEdit's 1.2M.

In addition, we also provide the results that trained with a lower resolution (256 $\times$ 256), the results on Real-Edit benchmark still outperforms previous SOTA method SmartEdit [24].

Method	Model Size Edit Data	Training Resolution	Following ↑ Acc	Following ↑ Score	Preserving ↑ Acc	Preserving ↑ Score	Quality ↑ Acc	Quality ↑ Score
SmartEdit	14.1B/1.2M	256	64%	3.50	66%	3.70	45%	3.56
SuperEdit	1.1B/40K	256	**68%**	**3.56**	**75%**	**4.02**	**66%**	**4.02**

Table 11. SuperEdit results with lower training resolution. Both SmartEdit and SuperEdit are pre-trained with InstructDiffusion here.

Novelty Claim. Our work introduces two key data-centric innovations to enhance image editing. First, we are the first to incorporate diffusion priors at the data preparation stage. While directly integrating noise-level-specific instructions into model training is a recognized yet unexplored challenge, our data-centric approach provides a solid and essential foundation for future advancements in this direction. Furthermore, we introduce a novel training methodology by leveraging negative prompts. Although these are widely used during inference, our key innovation lies in incorporating them directly into the training regimen through a contrastive triplet loss. This approach significantly improves the model's editing capabilities by explicitly teaching it to differentiate between desired outcomes and potential failure modes.

Limitation. Our method significantly enhances instruction-based image editing, but limitations still exist. The trained model still faces difficulties in understanding and executing complex instructions, especially with densely arranged objects and complicated spatial relationships. Although we used correction instructions and contrastive supervision signals, differences between editing results and editing instructions may still occur due to the inherent limitations of pre-trained Stable Diffusion and the challenges in fully capturing the nuances of natural language. Additionally, to fairly compare with existing methods, we chose Stable Diffusion v1.5 as the Base Model for building our editing model, which may result in worse image quality of edited images compared to state-of-the-art Text-to-Image models. Finally, ensuring the accuracy and effectiveness of correction and contrastive instructions requires the use of GPT-4o [1], which may incur additional costs as the amount of data increases.

13. More Visualization Comparison and Results

We show more visual comparison with existing instruction-based image editing methods in Fig. 12 and Fig. 13. Compared to existing instruction-based editing methods, our approach not only better understands and executes editing instructions but also preserves the original image's layout and quality more effectively, thereby significantly outperforming previous methods.

System Prompt for Instruction Rectification:

You are a professional image editor. I will give you two images later. The first image given is the original image, and the second is the edited image. You need to conduct a extremely detailed and step-by-step comparative analysis of the two input images according to the three independent aspects:
1. Overall Image Layout: Are there any changes in the composition and structure of the main content of the image, such as the number, size, focal length (zoom in/out), relative position, etc. of the main characters, main objects, and main background? Are there any entities that occupy a large space being deleted or added? In this section, please ignore the Texture, Motion, Pose, and Shape, Style, Color and Details.
2. Texture, Motion, Pose, and Shape: Are there any changes to the texture, motion, pose, or shape of the main characters, main objects, or main backgrounds? In this section, please ignore the Overall Image Layout, Style, Color and Details.
3. Style, Color and Details: Are there any changes to the color, tone, illumination, contrast, or style of all the object, background, or overall image? In this section, please ignore Overall Image Layout, and Texture, Motion, Pose, and Shape

When you write editing instructions, please follow these rules:
1. Describe the editing instructions directly without referring to the information of the input image. For example, "Change the clothes to red", do not output "Change the clothes from black to red".
2. Describe the changes clearly, for example, "Darker the lighting, change the colors to blue tones, and change the style to anime style", do not output "Adjust/change the lighting, color palette, and style".
3. Please describe only the parts that have been changed, and ignore the parts that have not been changed. For example, do not output “maintain/remains xxx”.

Then, please summarize and combine the analysis, clearly describe how to transform from the input image to the edited image. In the end, put the instructions in a Python dictionary in order and make sure the same format as the following. Python dicts can only be output once, and they should be put in the last.

```
Instruction = {
    "Overall Image Layout": "Detailed instruction",
    "Texture, Motion, Pose, and Shape": "Detailed instruction",
    "Style, Color and Details": "Detailed instruction",
    “Summarized Instruction”: “Combine and summarize the aforementioned details into a
      comprehensive and concise transformation guide."
}
```

System Prompt for Contrastive Instructions:

You are a professional image editor. I will give you two images. The first one is the original image, and the second one is the edited image. Then I will give you an editing instruction, which describes how to edit from the original image to the edited image. Now you need to change the correct input editing instructions to the wrong ones, including changing the quantity, position/relation, image style, color, category and attribute of the original editing instruction. Then integrate each modified editing instruction and return it in the form of a list. Please directly output the modified editing instructions in the following format:

```
Instruction = ["instruction with wrong quantity", "instruction with wrong position/relation", "instruction with wrong image style", "instruction with wrong color", "instruction with wrong object category", "instruction with wrong object attribute"]
```

Figure 9. GPT-4o prompts for constructing rectified and contrastive editing instructions

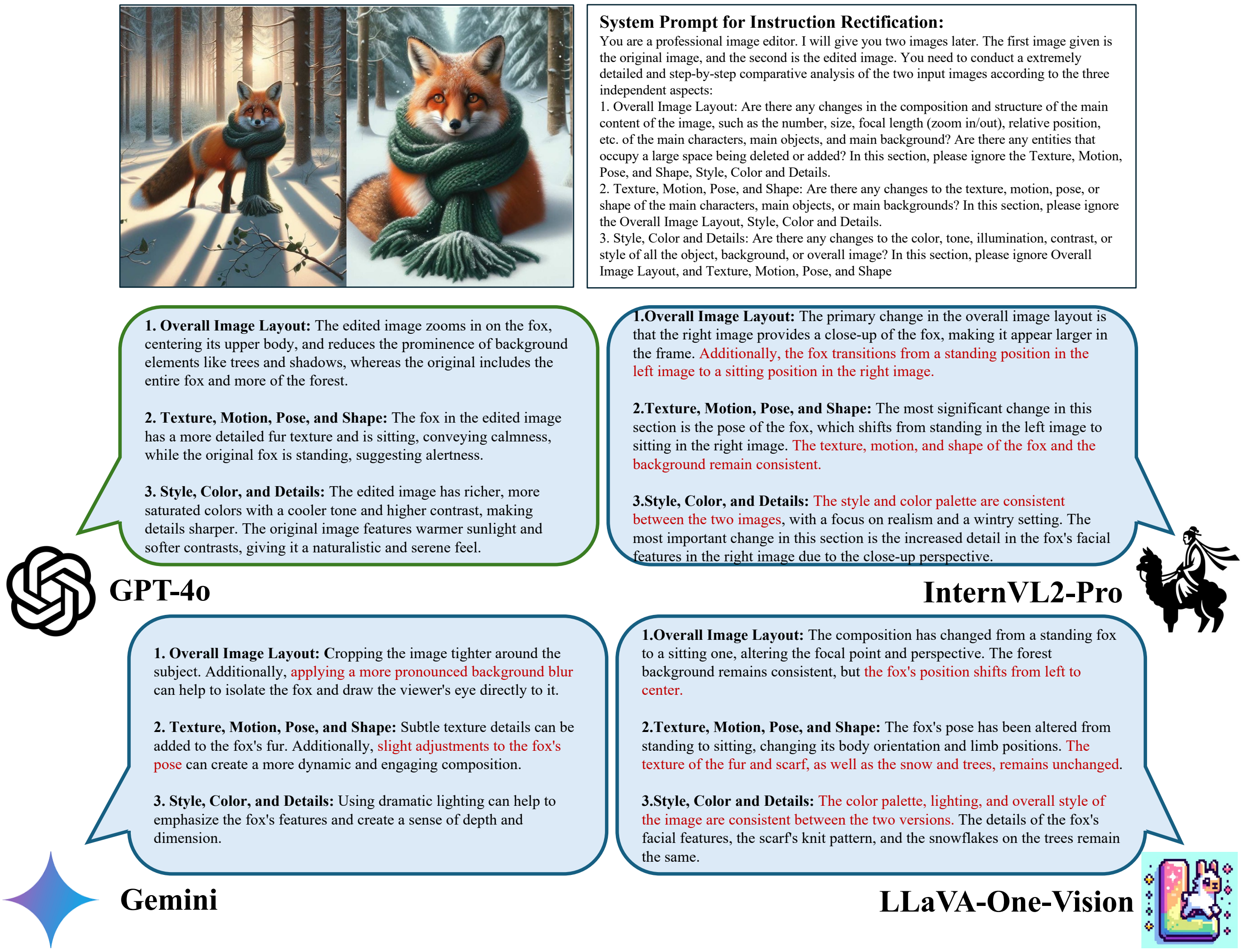

Figure 10. Comparison of different vision-language models in rectifying editing instructions based on generation prior attributes. GPT-4o achieves more stable and accurate results in describing the differences between original-edited image pairs. Text in red represents incorrectly generated instructions.

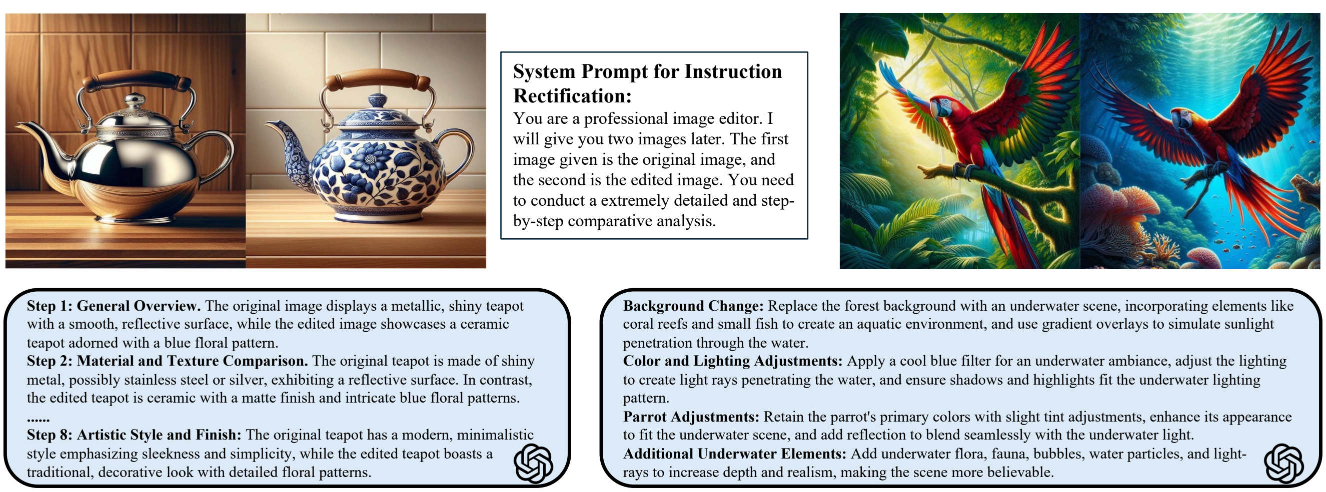

Figure 11. If the predefined four generation prior attributes are not used as templates for in-context learning, the GPT-4o rectified editing instructions will contain redundant information and lack the standardization needed for scalable processes.

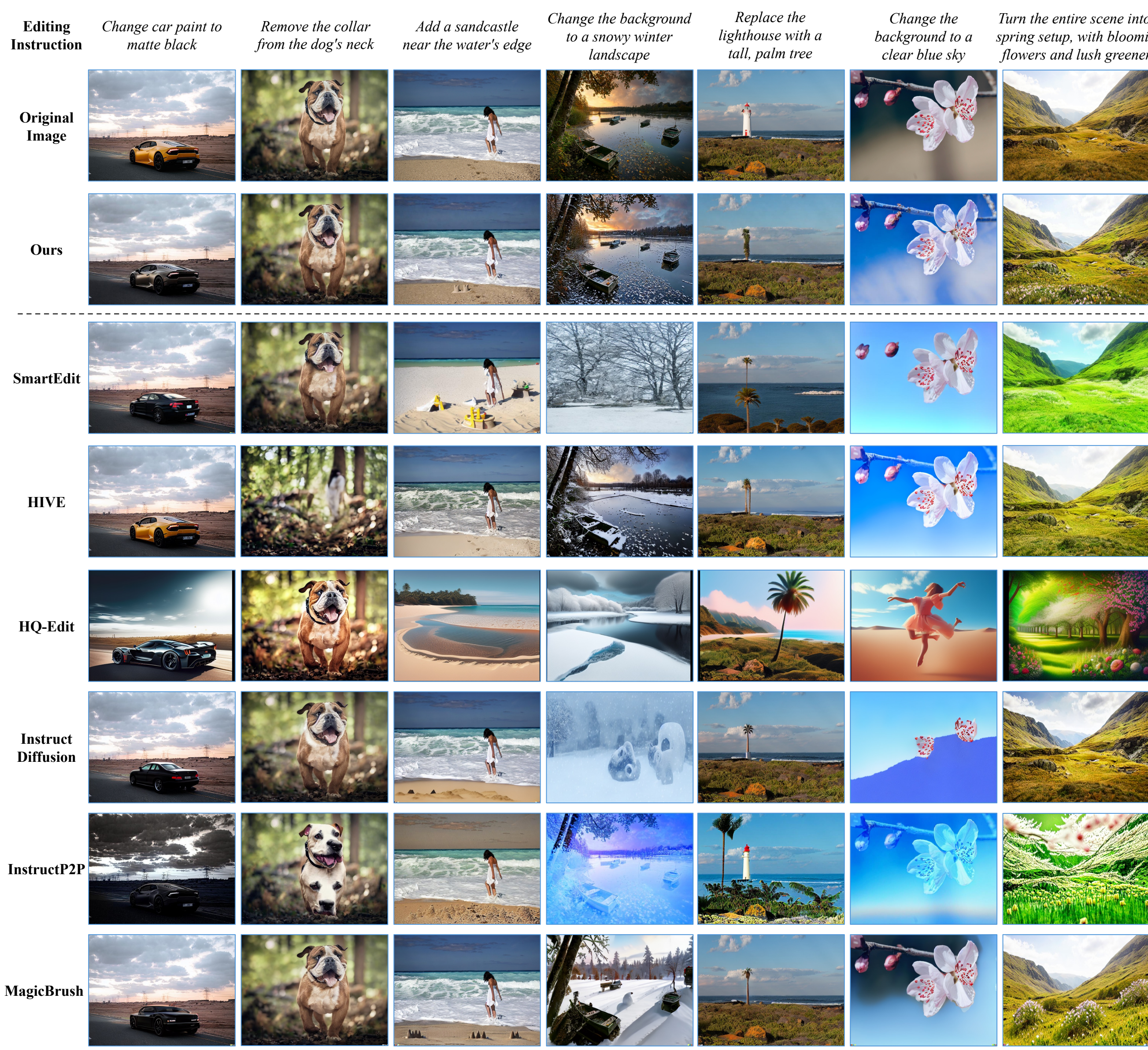

Figure 12. More visual comparison with existing methods.

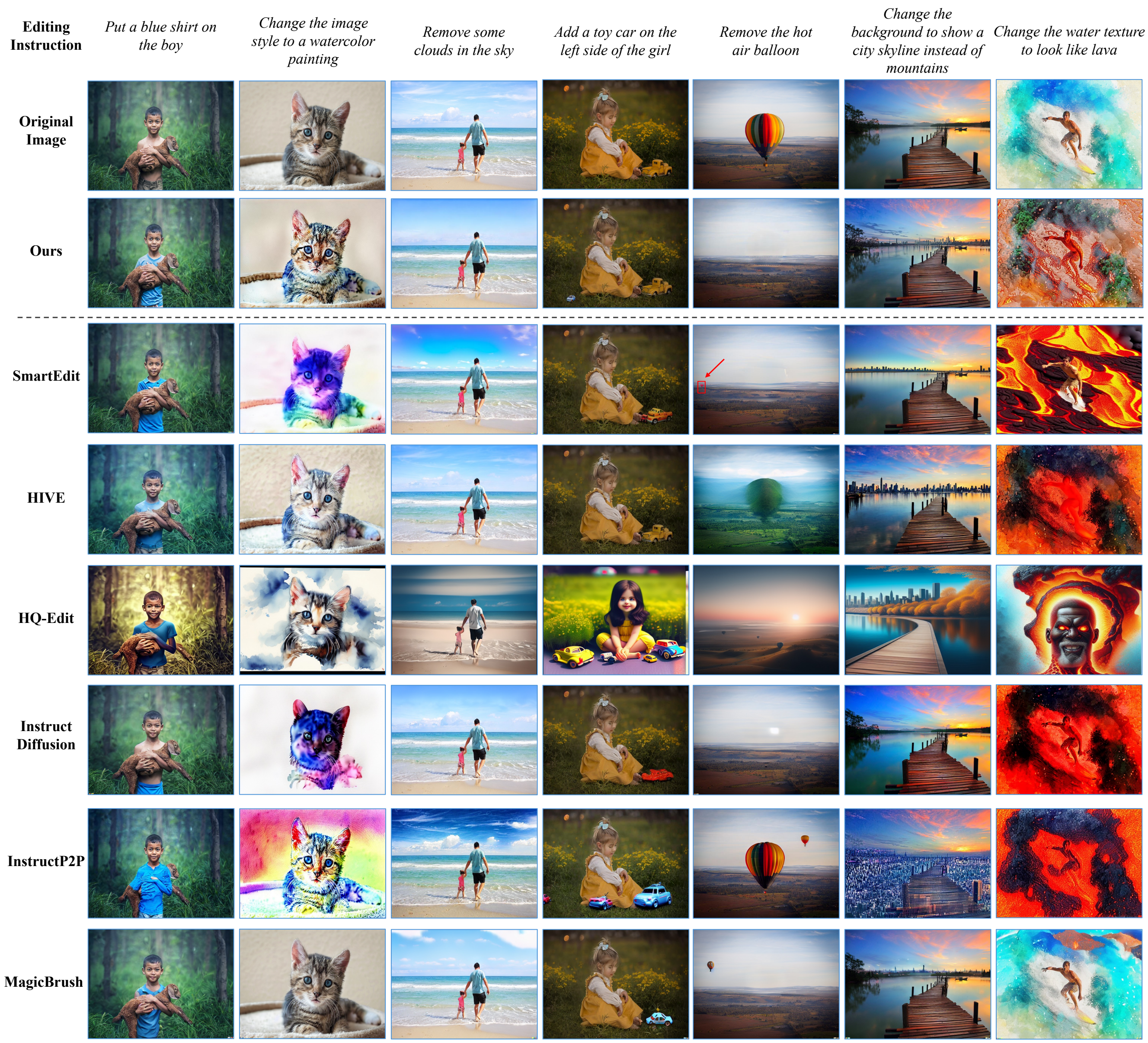

Figure 13. More visual comparison with existing methods.