Rethinking Real Image Editing: Unleashing Diverse Editing Operators via Multi-Objective Optimization

Supplementary Material

A. Algorithm Details

We introduce an inference-time multi-objective optimization algorithm to balance editing accuracy and content consistency for real image editing, which mainly consists of a reconstruction process and an editing process, as illustrated in Algorithm 1.

This algorithm leverages a pretrained diffusion model to denoise an image guided by the original prompt (y_{src}) during the reconstruction process. Importantly, it extracts and stores the internal representations of the image from each step of this denoising process, which provides the guidance of consistency during the editing process. Additionally, it employs the MGDA-UP algorithm during the editing process to generate a set of weights $\{v^1, \ldots, v^n\}$ to balance multiple objectives, ensuring all goals are optimized simultaneously. Finally, a reweighting strategy is designed to improve stability, which can dynamically balance the editing accuracy and consistency during editing.

B. Evaluation Protocol

Our proposed TEdBench-Opt includes a set of manually added, diverse editing operators, as shown in Tab. 4. In detail, we generate the original prompts using the BLIP model [26] and employ GPT-4o[31] to improve the alignment of BLIP-generated prompts with the target prompts. The prompt given to GPT-4o to refine the target prompts is as follows:

> “ Could you help me refine the target prompt for image editing tasks?
> There is an example.
> The original prompt is: ‘a dog standing in the grass staring ahead’, and the original target prompt is: ‘a photo of a sitting dog.’.
> The editing instruction is: ‘change action’.
> The answer is: ‘a dog sitting in the grass staring ahead.’
> Now, the question is to refine the new target prompt. The original prompt is {originalprompt}, and the target prompt is {targetprompt}. The editing instruction is {instruct}. Can you print the refined target prompt directly? ”.

This approach produces editing prompts that are closely aligned with the original prompts, making them more suitable for semantic image editing.

The PIE-bench encompasses nine distinct editing types, integrating both synthetically generated and authentically captured images. Our experiments filter 257 images from the PIE-bench. Our selection process prioritizes real images. Subsequently, we refined our image choices based on the associated textual prompts. Specifically, we excluded images accompanied by overly simplistic prompts and those pertaining to abstract conceptual editing, such as prompts involving “ghosts”. During this meticulous filtering phase, images exhibiting defocused facial features or indistinct primary subjects were systematically removed. Finally, we

Algorithm 1 An multi-objective optimization algorithm for real image editing

```
Require: $y_{src}$: the original prompt. $y_{tar}$: the target prompt. $\mathcal{O} = \{\mathcal{O}^1, \mathcal{O}^2, ..., \mathcal{O}^n\}$: editing operators. $n$: the number of editing operators. $\boldsymbol{\epsilon}_{\bar{\theta}}$: a pretrained diffusion model. $w$: the guidance scale of classifier-free guidance. $\phi$: the tokenizer. $\psi(\cdot)$: the text encoder. $O_\theta$: the word embeddings of the editing operators.
Ensure: $\mathbf{z}_0$: the final latent of the edited image.
  ▷Reconstruction.
  for $t = T, \ldots, 1$ do
      ▷ Extract representations.
      $\boldsymbol{\epsilon}^{rec}, I^* \leftarrow \boldsymbol{\epsilon}_{\bar{\theta}}(\mathbf{z}_t^*, \psi(\phi(y_{src})))$.
      ▷ Predicted noise to latent
      $\mathbf{z}_{t-1}^* \leftarrow UPDATE(\mathbf{z}_t^*, \boldsymbol{\epsilon}^{rec}, t)$
  end for
  ▷Editing with multi-objective optimization.
  for $t = T, \ldots, 1$ do
      ▷ Extract representations.
      $\boldsymbol{\epsilon}^{edi}, \mathcal{I} \leftarrow \boldsymbol{\epsilon}_{\bar{\theta}}(\mathbf{z}_t, \psi(\mathcal{O}_\theta, \phi(y_{tar})))$.
      ▷ Weights optimization
      for $i = 1, \ldots, n-1$ do
          $\mathcal{G}^i \leftarrow \nabla_{\mathbf{z}_t}\mathcal{L}^i(\mathcal{I}_i^*, \mathcal{I}_i)$.
      end for
      $\mathcal{G}^n \leftarrow \boldsymbol{\epsilon}^{edi} - \boldsymbol{\epsilon}^{rec}$.
      $\{v^i, \ldots, v^n\} \leftarrow$ MGDA-UP $(\mathcal{G}^i, \mathcal{G}^n)$.
      ▷ Reweight strategy
      $\gamma \leftarrow \frac{w\sum_{i=1}^{n-1} v^i}{\log_{10} v^n}$.
      ▷ Optimizing editing operators
      $\mathcal{O}_\theta^* \leftarrow \mathcal{O}_\theta - \gamma \sum_i^n v^i \cdot \nabla_{\mathcal{O}_\theta}\mathcal{L}^i(\mathcal{I}^*, \mathcal{I})$,
      ▷ Predicted noises during editing process
      $\boldsymbol{\epsilon}^* \leftarrow \boldsymbol{\epsilon}_{rec} + w(\boldsymbol{\epsilon}_{\bar{\theta}}(\mathbf{z}_t, \psi(\mathcal{O}_\theta^*, \phi(y_{tar}))) - \boldsymbol{\epsilon}_{rec})$.
      ▷ Predicted noise to latent
      $\mathbf{z}_{t-1} \leftarrow UPDATE(\mathbf{z}_t, \boldsymbol{\epsilon}^*, t)$
  end for
Return: $\mathbf{z}_0$
```

One-opt	change action	change object	change color	change background	add object
	change appearance	change status	change style	remove object	
Multi-opt	change object; change color		remove object; change status		change object; change object
	change background; add objects		change object; add object		change appearance; add object
	add object; add object; add object; add object				remove object; change object

Table 4. The editing operators contained in the TEdBench-Opt.

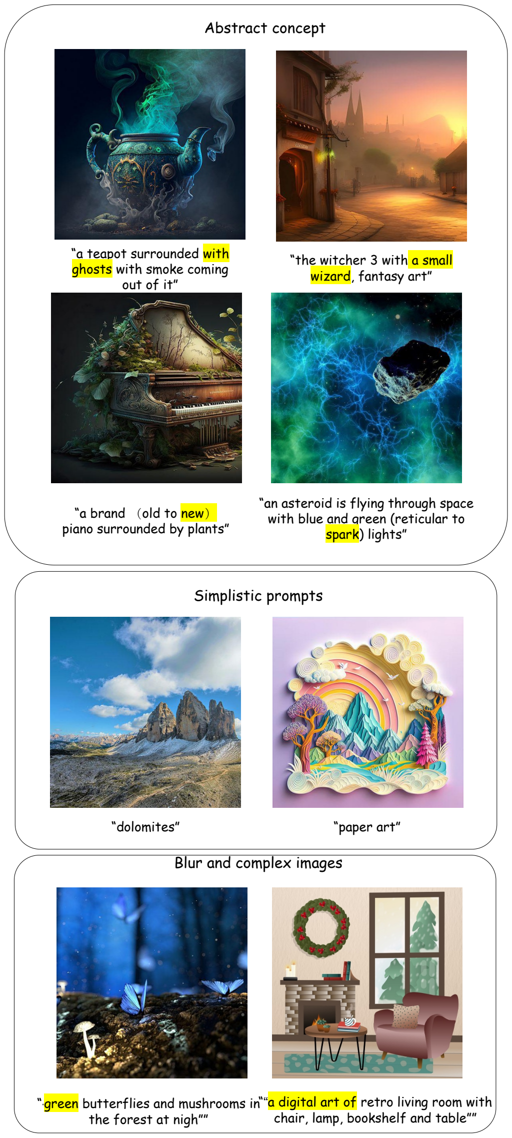

Abstract concept
"a teapot surrounded with ghosts with smoke coming out of it"
"the witcher 3 with a small wizard, fantasy art"
"a brand (old to new) piano surrounded by plants"
"an asteroid is flying through space with blue and green (reticular to spark) lights"
Simplistic prompts
"dolomites"
"paper art"
Blur and complex images
"green butterflies and mushrooms in "the forest at nigh""
""a digital art of retro living room with chair, lamp, bookshelf and table""

Figure 6. The examples of filtered images in PIE-bench

eliminated complex indoor scenes, particularly those containing multiple discernible objects. Highly cluttered scenes can introduce numerous unforeseen challenges for editing methods, making it difficult to attribute successes or failures to specific editing operations. This simplification allows for a more controlled and interpretable evaluation environment, enabling clearer insights into the image editing methods' strengths and weaknesses. The examples of filtered images are shown in Fig. 6. We will release the selected PIE-bench upon acceptance.

C. Evaluation Metrics

Following our baselines [2, 33], we quantify the performance of our method and baselines with three traditional metrics: 1) CLIPScore [15], assessing the semantic alignment of edited images with the target prompts in CLIP latent space; 2) Structure Dist (Str Dist) [41], evaluating the structural consistency between the real images and the edited images; and 3) LPIPS [51], measuring the pixel-level similarity of the real to the edited image.

Traditional metrics for image editing fail to accurately evaluate two critical aspects: editing accuracy and the ability to maintain consistency during the editing process, as shown in Fig. 8. On the one hand, CLIPScore struggles to assess the accuracy of fine-grained edits, such as changes in an object's spatial relationships or dynamic actions, which are also mentioned in previous evaluation methods [8, 32, 49]. For instance, as shown in the left part of Fig. 7, despite our method successfully changing the dog's action, it receives a lower score than a baseline that fails to perform the edit, which demonstrates that CLIPScore is flawed in evaluating editing accuracy in editing tasks. Previous methods discuss the text-to-image synthesis evaluation via GPT-4 [8] as a judge and human evaluations [32, 49]. However, there is a scarcity of evaluation metrics tailored to image editing.

On the other hand, for editing consistency metrics [41, 51], such as Str Dist and LPIPS, those metrics evaluate consistency by comparing the edited image to the original without accounting for the specific editing task. This leads to a misleading result, as many methods achieve high scores by simply reconstructing the original image rather than performing an edit. As shown in the right part in Fig. 7, most baselines, such as Stable Flow [1], Ledit++ [2], and Guide-and-Rescale [40], achieve superior consistency scores by sacrificing editing effectiveness. This demonstrates that these metrics fail to truly reflect a method's ability to maintain consistency during an editing process, especially for controllable image editing. These limitations in traditional metrics motivated the newly designed GPT-based metrics.

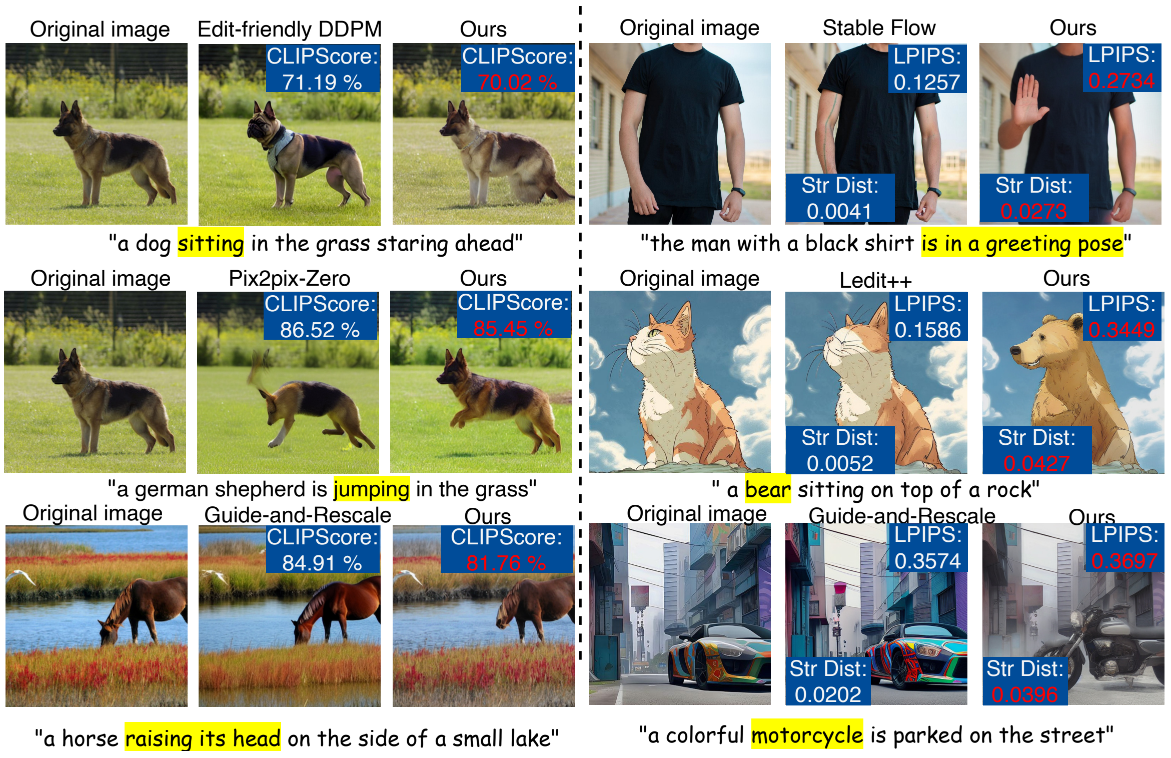
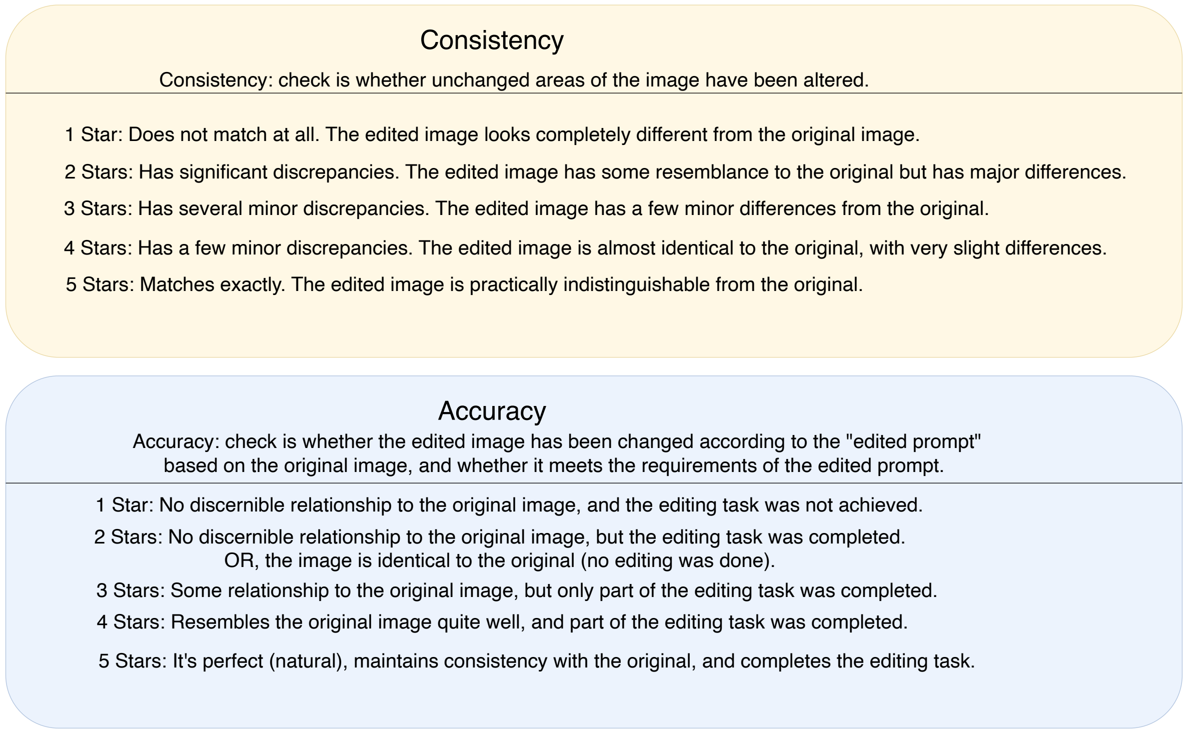


Figure 7. The examples of limitations in existing metrics. Left: CLIPScore fails to evaluate fine-grained edits; our method gets a lower score than a baseline despite a successful edit. Right: most baselines achieve a better consistency score but ignore the edits.



Consistency

Consistency: check is whether unchanged areas of the image have been altered.

1 Star: Does not match at all. The edited image looks completely different from the original image.

2 Stars: Has significant discrepancies. The edited image has some resemblance to the original but has major differences.

3 Stars: Has several minor discrepancies. The edited image has a few minor differences from the original.

4 Stars: Has a few minor discrepancies. The edited image is almost identical to the original, with very slight differences.

5 Stars: Matches exactly. The edited image is practically indistinguishable from the original.

Accuracy

Accuracy: check is whether the edited image has been changed according to the "edited prompt" based on the original image, and whether it meets the requirements of the edited prompt.

1 Star: No discernible relationship to the original image, and the editing task was not achieved.

2 Stars: No discernible relationship to the original image, but the editing task was completed.
OR, the image is identical to the original (no editing was done).

3 Stars: Some relationship to the original image, but only part of the editing task was completed.

4 Stars: Resembles the original image quite well, and part of the editing task was completed.

5 Stars: It's perfect (natural), maintains consistency with the original, and completes the editing task.

Figure 8. The human-evaluation metrics for editing accuracy and consistency with original images.

To evaluate the generated images using GPT-4o, we categorize the editing prompts based on five aspects: object, action, attribute, background, and style. We use GPT-4o to perform this categorization. The prompt designed to identify the editing information for each editing example is as follows:

" I currently need to extract some keywords from the editing prompt, covering five aspects: Object, Action, Attribute, Background, and Style.
There are some default rules:
{{
If the editing prompt does not contain style information, the default is " photo ".
If the editing prompt does not contain Action, the default is None.
If the editing prompt does not contain Attribute, the default is None.
If the editing prompt does not contain Background, the default is None.
If the editing prompt contains no or nothing, the default editing prompt is {oldprompts+"and"+editedprompt}.
}}
Here is an example:
editing prompt: " a black and brow dog sitting in the grass staring ahead."
The returned JSON is:
{{
Object: dog,
Action: sitting,
Attribute: black and brow,
Background: in the grass,
Style: photo,
}}
Here is an example of the editing prompt that contains no or nothing:
editing prompt: the white horse is standing in the sun, facing far away and nothing is standing in the sun, facing far away.
The returned JSON is:
{{
Object: no white horse,
Action: no standing,
Attribute: None,
Background: in the sun,
Style: photo,
}}
Please generate the JSON output based on the above requirements and example.
Edited prompt:{editedprompt}"

Following the identification of the target prompts, we obtain editing details: object, action, attribute, background, and style. Based on this information, we can assess the accuracy of the editing. The editing results are considered accurate (Acc(G)=1) only if all editing intents are 1. Additionally, we also intend to evaluate the structure and appearance consistency with the original image. Likewise, consistency (Con(G)=1) is achieved only when both structural and appearance consistency are evaluated as successful. To generate the quality scores, we use the following prompt:

" I am currently working on an image editing task, which requires modifying an image according to a target prompt while preserving the original image as much as possible.
My input is:
{{
original prompt: {oldprompts[i]}, which represents the original image corresponding to the original prompt.
editing prompt: {newprompts[i]}, which means the edited image needs to be modified from the original image to match this target prompt. }}
I will provide two images:
{{
The first image is the original image.
The second image is the edited image.
}}
Your task is to evaluate the quality of the editing based on the following criteria:
{{
1. Object: The expected object on the edited image is {objs[i]}. If the object is None, automatically return 1. Otherwise, check if the edited image contains this object and whether it has been edited successfully.If yes, return 1; otherwise, return 0.
2. Action: The expected action on the edited image is {acts[i]}. If the action is None, automatically return 1. Otherwise, check if the object in the edited image is performing the specified action. If yes, return 1; otherwise, return 0.
3. Attribute: The expected attribute on the edited image is {atts[i]}. If the attribute is None, automatically return 1. Otherwise, check if the object in the edited image has this attribute. If yes, return 1; otherwise, return 0.
4. Background: The expected background on the edited image is {backs[i]}. If the background is None, automatically return 1. Otherwise, check if the background of the {objs[i]} on the edited image is backs[i]. If yes, return 1; otherwise, return 0.

5. Style: The expected style of the edited image is styles[i]. Check if the edited image retains a styles[i] style. If yes, return 1; otherwise, return 0.
6. Structural Consistency: Evaluate whether the non-edited regions in the edited image maintain structural consistency with the original image. Only check if the non-edited regions in the edited image retain shape, position, and other structural information as in the original image. If yes, return 1; otherwise, return 0.
7. Pixel-level Consistency: Evaluate whether the non-edited regions in the edited image maintain pixel-level consistency with the original image. If yes, return 1; otherwise, return 0.
}}
Please return the evaluation results in JSON format, for example:
{{
Object: 0,
Action: 1,
Attribute: 0,
Background:1,
Style: 1,
StrConsistency: 1,
PixConsistency: 1
}}
First, provide an explanation of your assessment. Then, output the evaluation results in JSON format."

This methodology assigns a binary score to each of the following assessment intents: object, action, attribute, background, style, structural consistency, and pixel-level consistency. These scores provide a comprehensive evaluation metric for controllable image editing tasks, considering editing accuracy across the five editing detail intents (object, action, attribute, background, style), and consistency across both structural and pixel levels. The overall success rate (SR(G)) delivers a rigorous and comprehensive measure of overall editing quality.

We conducted a user study, as detailed in Sec. 4.3.3 of the main paper. Users were tasked with evaluating the accuracy and consistency of edited images, based on the criteria outlined in Fig. 8. Additionally, users rated the baselines and our method as shown in Fig. 9. Each star represents a rating point, with higher star ratings indicating a more successful edit.

D. Implementation Details

Our experiments were conducted using the Stable Diffusion 1.5 model, implemented with version 0.27.0 of the diffusers library. For reproducibility, the random seed was fixed at 42, and the inference step was set to 50. Our text inputs consist of editing prompts and editing operators. Our textural inputs consist of editing prompts and editing operators. According to the diversity of images and editing tasks, we adjusted the guidance scale (w) for classifier-free guidance. We employ guidance scales of {7.5, 10, 15} for editing, learning rates for consistency guidance of {0.1, 0.001, 0.008, 0.005}, and editing steps of {20, 30, 40, 50}. For each image, the optimal result was selected from these hyperparameter configurations as the final output.

The implementation details for the baseline models are described below:

1) **SDEdit** [28]: was implemented using the diffusers package, specifically the Stable Diffusion img2img pipeline (https://huggingface.co/docs/diffusers/en/api/pipelines/stable_diffusion/img2img). Experiments were conducted on Stable Diffusion v1.5 with its default settings. The process used both original and editing text prompts to guide image generation and modification.

2) **Pix2pix-Zero** [33]: We implemented it based on the official Pix2pix-Zero project (https://github.com/pix2pixzero/pix2pix-zero) utilizing Stable Diffusion v1.5 with its default parameters, such as a learning rate between 0.1 and 0.2. Our modification involved using refined target prompts as inputs to the original Pix2pix-Zero implementation.

3) **Edit-friendly DDPM** [19]: Our implementation utilized the official project from (https://github.com/inbarhub/DDPM_inversion) with default parameters. Experiments were conducted using Stable Diffusion v1.5, configured with its default settings. For textual guidance, original prompts and editing prompts were provided as text inputs.

4) **Ledit++** [2]: We utilized the edited images provided by Ledit++ on the TEdBench++ dataset, accessible via (https://huggingface.co/datasets/AIML-TUDA/TEdBench_plusplus), which manually adjusted parameters for each image. For the PIE-bench evaluation, we utilized the official Ledit++ code from (https://github.com/ml-research/ledits_pp) with default parameters. During the editing process, we defined the original prompts as "removed concepts" and the edited prompts as "added concepts".

5) **Guide-and-Rescale** [40]: Using the default parameters of the Guide-and-Rescale implementation (available at https://github.com/AIRIInstitute/Guideand-Rescale/tree/main), we performed experiments on Stable Diffusion v1.5. Both original and edited text prompts served as guidance during the process.

6) **Stable Flow** [1]: We implemented the flow-based method using FLUX.1-dev [24]. Using the default parameters of the Stable Flow implementation (available at https://github.com/snap-research/stable-flow), we performed various types of image editing with original and edited text prompts.

7) **Plug-and-play** (PnP) [42]: We implemented it us-

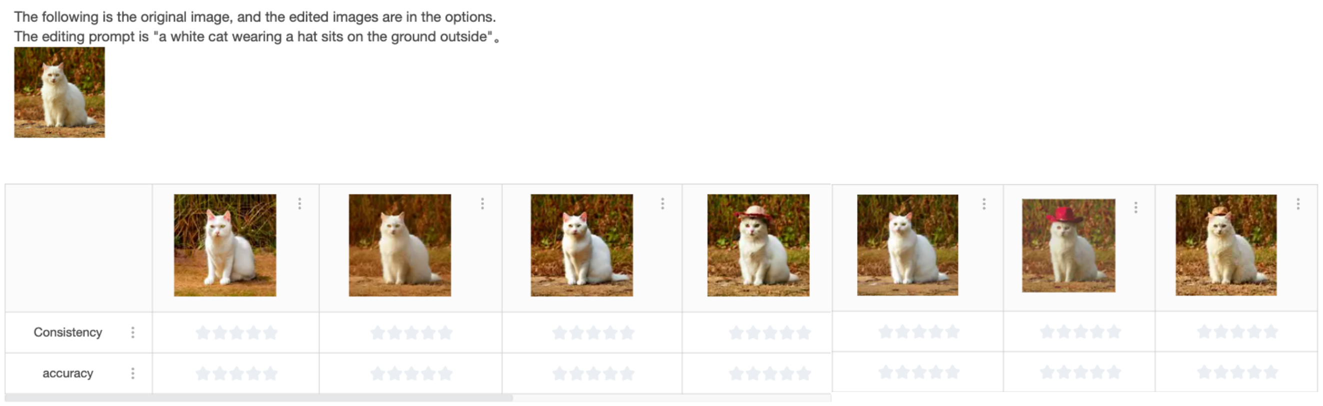

The following is the original image, and the edited images are in the options.
The editing prompt is "a white cat wearing a hat sits on the ground outside"。

Figure 9. An example of user study on consistency and accuracy evaluation.

SDEdit	Pix2pix-Zero	Edit-friendly DDPM	Ledit++	Guide-and-Rescale	Stable Flow	Plug-and-play	Directpix2pix	Masactrl	Null-text Inversion	Ours
13s	39s	28s	17s	98s	137s	27s	55s	28s	93s	107s

Table 5. Inference-time comparisons of our method with baselines.

ing Stable Diffusion v1.5, based on the official PnP Diffusion project (https://github.com/MichalGeyer/plug-and-play?tab=readme-ov-file) with its default settings. We defined the original and edited prompts as inputs.

8) **Directpix2pix** [21, 33]: is implemented by utilizing Direct [21] as inversion method and pix2pix-zero [33] as editing method. We implemented the inversion-based method using Stable Diffusion v1.5, based on the official Direct inversion project (https://github.com/cure-lab/PnPInversion) with default parameters. The inputs are original and edited text prompts.

9) **Masactrl** [4]: We implemented MasaCtrl using Stable Diffusion v1.5, based on its official project (https://github.com/TencentARC/MasaCtrl) with its default parameters. The inputs comprised an original and a target text prompt.

10) **Null-text Inversion** (Null-text) [30]: Our implementation of Null-text Inversion utilized Stable Diffusion v1.5, based on its official project (https://github.com/google/prompt-to-prompt). During the inversion process, the inputs consisted of an original image and its corresponding original text prompt. During the editing process, the inputs were the null-text inversion optimized unconditional text embedding and the edited prompt.

11) **InstructPix2pix** [3]: We implemented InstructPix2pix using Stable Diffusion v1.5, leveraging the diffusers package and the implementation code from its official project (https://github.com/timothybrooks/instruct-pix2pix). We used the original image, edited prompts, and editing instructions as inputs.

Method	CLIPScore (↑)	Str Dist (↓)	LPIPS (↓)
SDEdit [28]	77.65%	0.0665	0.4968
Pix2pix-Zero [33]	79.83%	0.0288	0.2987
Edit-friendly DDPM [19]	80.50%	0.0212	0.2640
Ledit++ [2]	78.23%	0.0395	0.3670
Guide-and-Rescale [40]	80.45%	0.0220	0.2930
Stable Flow [1]	77.67%	0.0143	0.1668
Ours	78.82%	0.0341	0.3343

Table 6. Quantitative comparison of our method against baseline approaches on traditional metrics.

Method (G)	SR (↑)	ES (↑)	Acc (↑)	Con (↑)
PnP [42]	29.60%	**85.93%**	35.43%	**78.70%**
Directpix2pix [21, 33]	41.73%	85.61%	53.54%	72.44%
Masactrl [4]	22.83%	81.10%	33.07%	66.92%
Null-text [30]	13.38%	69.06%	51.96%	18.89%
InstructPix2pix [3]	14.17%	82.56%	51.69%	18.11%
Ours	**48.81%**	85.82%	**57.48%**	71.65%

Table 7. More quantitative comparisons on TEdBench-Opt

E. Inference-time Comparisons

As shown in the Tab. 5, we evaluate the inference time by averaging the duration of 250 individual image edits. All methods are evaluated on a single NVIDIA A6000 GPU. Our inference time is 107s, which is slightly slower than many non-optimization-based approaches. Unlike baselines optimized for a single objective and/or a single task, our method simultaneously balances accuracy and consistency across a wide range of edits. Thus, the increased time is a necessary trade-off for our method's superior versatility and performance.

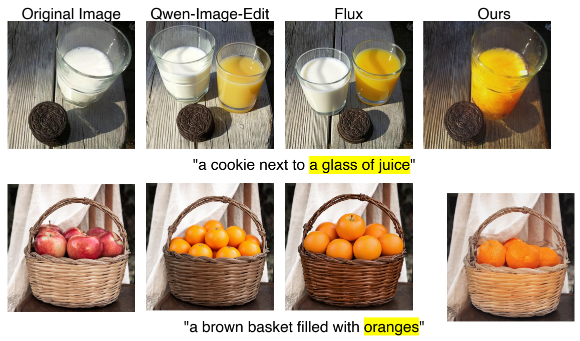

Original Image
Qwen-Image-Edit
Flux
Ours
"a cookie next to a glass of juice"
"a brown basket filled with oranges"

Figure 10. Qualitative comparison of our method with train-based SOTA method.

F. More Quantitative Comparisons

Traditional evaluations are conducted on TEdBench-Opt [2] to compare our method with prior methods, as illustrated in Tab. 6. Compared with baseline models, although some baseline methods achieve higher CLIPScore [15], this traditional metric does not directly reflect editing accuracy, which is also discussed in previous evaluation methods [8, 32, 49]. Additionally, traditional consistency evaluation metrics, such as LPIPS [51] and Str Dist [41], are limited, only evaluating global consistency and failing to differentiate edited from non-edited regions. Moreover, different editing tasks demand various-grained consistency. For example, a style transfer task may primarily require evaluating structural consistency, while the addition of an object may require localized structure adjustments. Therefore, a lower LPIPS or Structure Dist score may not always indicate higher editing quality. While these traditional metrics may not fully capture editing quality and can be imprecise, our method still achieves commendable scores across them, further demonstrating its effectiveness in terms of image accuracy and consistency with the original image.

The quantitative comparisons with more baselines on TEdBench-Opt, detailed in Tab. 7, indicate that most baselines struggle to balance consistency and accuracy (lower SR) and only optimize one objective. For instance, PnP achieves a high consistency score with original images (Con (G), but suffers from low accuracy (Acc (G)) and a significantly lower successful rate (SR (G)). This suggests that PnP effectively preserves the overall structure but often fails to precisely execute the desired edits. Conversely, InstructPix2pix show higher accuracy (Acc (G)), implying they are better at editing images with text guidance. However, they often exhibit a noticeable drop in consistency with the original image (Con (G)). Overall, our proposed method consistently outperforms these baselines in editing quality by effectively balancing both editing accuracy and consistency.

Method (G)	SR (↑)	ES (↑)	Acc (↑)	Con (↑)
Qwen-Image-Edit [47]	46.4%	85.7%	**82.6%**	53.5%
Flux-Kontext [25]	25.1%	57.8%	29.9%	44.8%
Stable Flow [1]	28.3%	**86.7%**	32.2%	**84.2%**
Ours	**48.8%**	85.8%	57.4%	71.6%

Table 8. More quantitative comparisons of our method with flow-based state-of-the-art methods on TEdBench-Opt.

G. More Qualitative Results

We present more comprehensive qualitative comparisons of our proposed method against established baselines on both the TEdBench-Opt and PIE-bench datasets.

In Fig. 11, Fig. 12, and Fig. 13, we showcase the superior performance of our approach on the TEdBench-Opt dataset. Our method consistently surpasses baselines—both traditional techniques and more recent advancements—across a diverse array of editing tasks, including style transfer, background replacement, appearance modification, action transformation, and color manipulation. Conclusively, our results consistently demonstrate an excellent balance between maintaining consistency with the original image and effectively executing the editing intentions, leading to an overall robust performance.

Further qualitative results on the PIE-bench dataset are depicted in Fig. 14. While Ledit++ [2] excels in tasks involving object alteration and addition, it exhibits identifiable weaknesses in executing background changes and style transfer operations. This observation aligns with its comparatively high consistency score on PIE-bench, as detailed in Tab. 1 in the main paper, indicating a tendency to prioritize image preservation over edits on PIE-bench. Conversely, methods such as Edit-friendly DDPM [19] and Guide-and-Rescale [40], achieve desired edits with the cost of maintaining high consistency with the original image. Stable Flow [1] often preserves a high degree of fidelity to the original images, without effectively conducting the intended editing directives.

Collectively, these qualitative analyses underscore a critical advantage of our method: it consistently achieves the specified editing intentions while robustly preserving the consistency with the original input images. This dual strength highlights our method's efficacy and practical utility across diverse image editing tasks.

H. Comparisons with Flow-based Methods

As shown in the Tab. 8 and Fig. 10, we conduct further quantitative and qualitative comparisons of our method with flow-based state-of-the-art models on the TEdBench-Opt dataset. Our method achieved a high success rate, demonstrating superior performance in completing editing tasks while preserving the overall coherence and integrity of the

	w=7.5				w=10				w=15			
	SR	ES	Acc	Con	SR	ES	Acc	Con	SR	ES	Acc	Con
lr=0.1	15.7%	77.50%	25.9%	65.3%	15.7%	77.50%	29.9%	63.7%	7.8%	65.46%	22.8%	39.3%
lr=0.008	20.4%	79.86%	35.4%	64.5%	21.2%	78.51%	34.6%	58.2%	11.8%	71.31%	28.3%	40.9%

Table 9. Hyperparameter sensitive analysis.

image. In contrast, training-based baselines often struggle with maintaining consistency during the editing process. For instance, Qwen-Image-Edit fails to change the milk to juice, instead adding a glass of juice beside the original. Similarly, Flux ignores the original structure and the number of apples. These examples reveal that while these methods can perform edits, they may struggle to preserve the consistency with the original image.

I. Hyperparameter Sensitive Analysis

Our method's performance is sensitive to key hyperparameters, particularly the Classifier-free guidance scale (w) and the learning rate for consistency control (lr), as shown in Tab. 9. As w increases, the edited image relies more heavily on text guidance, which can compromise its adherence to the original image's content. This degrades the method's editing capability for subtle tasks, such as minor changes. Furthermore, increasing the lr for consistency control improves image consistency, as evidenced by our experiments at $w = 7.5$ and $w = 10$, where a higher lr of 0.1 shows better consistency. However, this comes at a significant cost to editing accuracy and success rate. Therefore, we find that optimal parameter settings depend on the complexity of the editing task. For difficult tasks involving large editing regions, increasing w is beneficial. Conversely, for simpler edits with smaller changes, the lr should be reduced to maintain a high degree of editing accuracy. In practice, users can fine-tune these two hyperparameters to match their specific editing tasks.

J. Discussions

We propose Remo, an inference-time real-image editing framework that enables multiple editing tasks through a novel multi-objective optimization strategy. The strong generalization of our approach stems from two key factors: its leverage of a large-scale pre-trained diffusion model and its inference-time tuning strategy. To validate this, we conducted extensive qualitative evaluations on PIE-bench and TEdBench-Opt, which contain nine editing types and more than 30 objects. As shown in Fig. 15, our method successfully performs edits across a wide range of scenarios, including intricate indoor and outdoor scenes, edits on various animals, food, humans, and even anime characters. This ability to quickly adapt to new tasks by tuning the word embeddings of our editing operators is what enables our method to generalize across such diverse editing tasks.

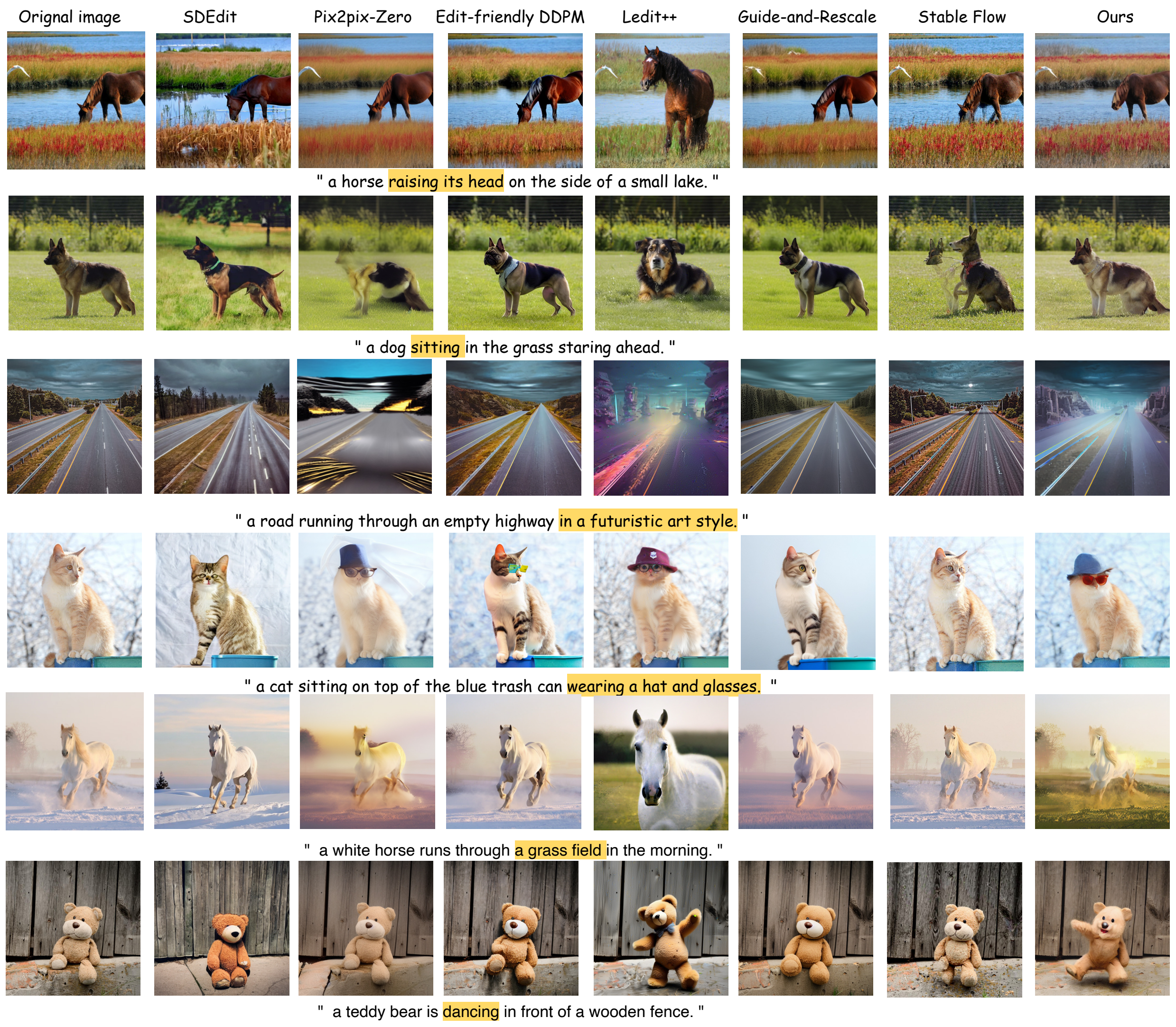

Orignal image
SDEdit
Pix2pix-Zero
Edit-friendly DDPM
Ledit++
Guide-and-Rescale
Stable Flow
Ours
" a horse raising its head on the side of a small lake. "
" a dog sitting in the grass staring ahead. "
" a road running through an empty highway in a futuristic art style. "
" a cat sitting on top of the blue trash can wearing a hat and glasses. "
" a white horse runs through a grass field in the morning. "
" a teddy bear is dancing in front of a wooden fence. "

Figure 11. Qualitative comparison of our method with baselines on various editing tasks (changing actions, style transfer, changing backgrounds, and appearances). Our method achieves higher editing accuracy while maintaining consistency with the original images.

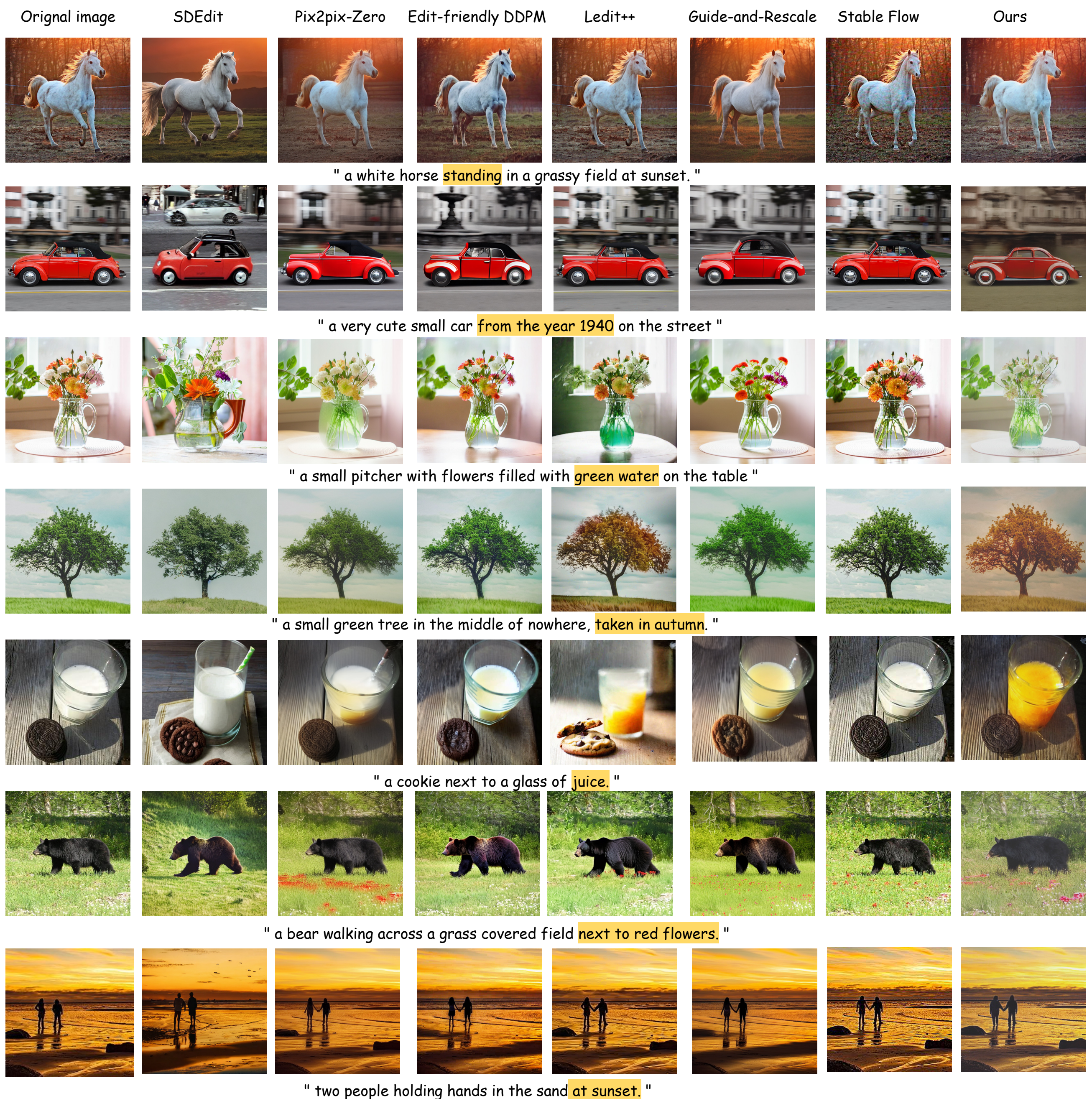

Orignal image
SDEdit
Pix2pix-Zero
Edit-friendly DDPM
Ledit++
Guide-and-Rescale
Stable Flow
Ours
" a white horse standing in a grassy field at sunset. "
" a very cute small car from the year 1940 on the street "
" a small pitcher with flowers filled with green water on the table "
" a small green tree in the middle of nowhere, taken in autumn. "
" a cookie next to a glass of juice. "
" a bear walking across a grass covered field next to red flowers. "
" two people holding hands in the sand at sunset. "

Figure 12. Qualitative comparison of our method with baselines on various editing tasks, including changing actions, style transfer, color adjustment, object editing, and background replacement. Our method achieves higher editing accuracy while maintaining consistency with the original images. Our method demonstrates superior accuracy and consistency with the original images.

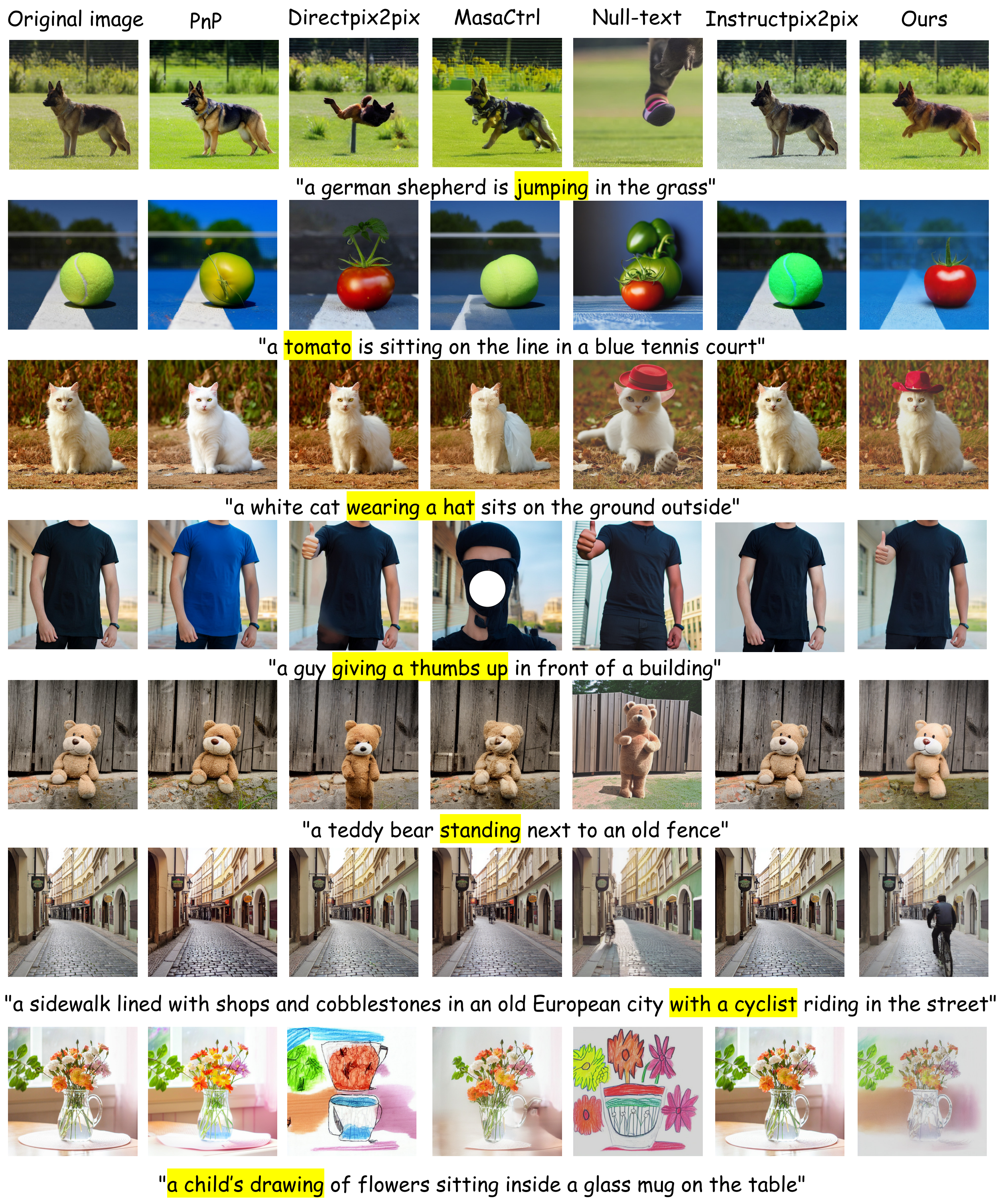

Original image
PnP
Directpix2pix
MasaCtrl
Null-text
Instructpix2pix
Ours
"a german shepherd is jumping in the grass"
"a tomato is sitting on the line in a blue tennis court"
"a white cat wearing a hat sits on the ground outside"
"a guy giving a thumbs up in front of a building"
"a teddy bear standing next to an old fence"
"a sidewalk lined with shops and cobblestones in an old European city with a cyclist riding in the street"
"a child's drawing of flowers sitting inside a glass mug on the table"

Figure 13. More qualitative comparison of our method with baselines on various editing tasks, including changing actions, changing objects, adding objects, and style transfer. Our method achieves superior editing accuracy while preserving consistency with the original images.

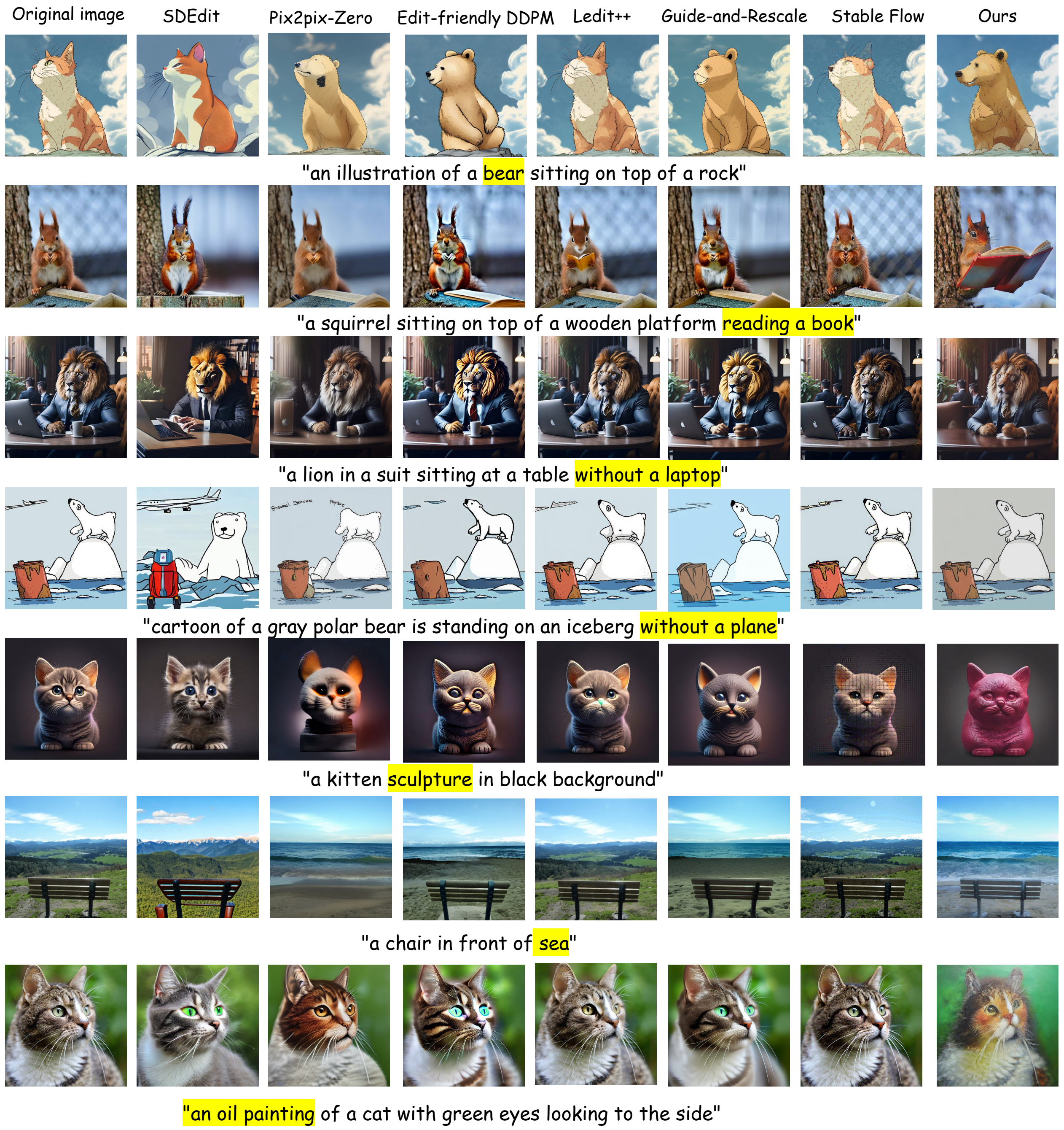

Original image
SDEdit
Pix2pix-Zero
Edit-friendly DDPM
Ledit++
Guide-and-Rescale
Stable Flow
Ours
"an illustration of a bear sitting on top of a rock"
"a squirrel sitting on top of a wooden platform reading a book"
"a lion in a suit sitting at a table without a laptop"
"cartoon of a gray polar bear is standing on an iceberg without a plane"
"a kitten sculpture in black background"
"a chair in front of sea"
"an oil painting of a cat with green eyes looking to the side"

Figure 14. More qualitative comparison of our method with baselines on PIE-bench, including changing objects, adding objects, removing objects, changing attributes, editing background and style transfer. Our method achieves superior editing accuracy while preserving consistency with the original images.

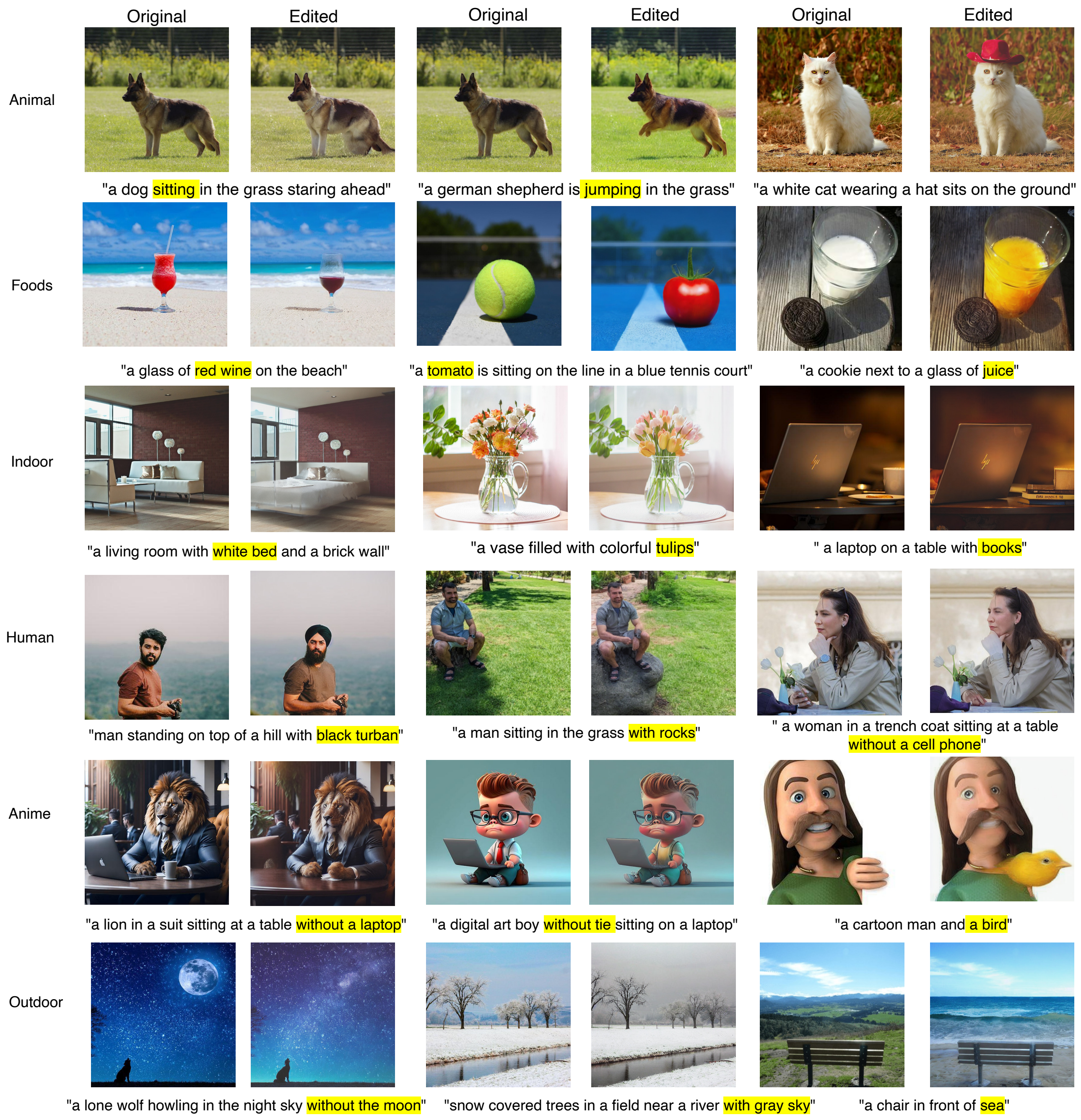

Original
Edited
Original
Edited
Original
Edited
Animal
"a dog sitting in the grass staring ahead"
"a german shepherd is jumping in the grass"
"a white cat wearing a hat sits on the ground"
Foods
"a glass of red wine on the beach"
"a tomato is sitting on the line in a blue tennis court"
"a cookie next to a glass of juice"
Indoor
"a living room with white bed and a brick wall"
"a vase filled with colorful tulips"
" a laptop on a table with books"
Human
"man standing on top of a hill with black turban"
"a man sitting in the grass with rocks"
" a woman in a trench coat sitting at a table without a cell phone"
Anime
"a lion in a suit sitting at a table without a laptop"
"a digital art boy without tie sitting on a laptop"
"a cartoon man and a bird"
Outdoor
"a lone wolf howling in the night sky without the moon"
"snow covered trees in a field near a river with gray sky"
"a chair in front of sea"

Figure 15. More qualitative results of our method on various image editing tasks. Our approach successfully edits diverse scenarios, including editing animals, changing objects like food, modifying indoor and outdoor scenes, and editing human and anime images.