Paint by Inpaint: Learning to Add Image Objects by Removing Them First

Supplementary Material

A. Additional Model Outputs

In continuation of the demonstrations seen in Figure 1, we further show a variety of object additions performed by our model in Figure S8. The editing results showcase the model's ability to not only add a diverse assortment of objects and object types but also to integrate them seamlessly into images, ensuring the images remain natural and appealing. Additionally, in Figure S9, we provide an example of our model's capability to generate diverse results for the same edit using different seeds.

B. General Editing

As detailed in Section 6, the model, trained on the combined IP2P and PIPE dataset, achieves new state-of-the-art scores for the general editing task. In Figure S11, we present a visual comparison that contrasts our model's performance with that of a model trained without the PIPE dataset. The results not only underscore our model's superiority in object additions but also demonstrate its effectiveness in enhancing outcomes for other complex tasks, such as object replacement.

We further analyze this model by testing its performance not on the entire MagicBrush dataset as in Section 6, but on the 'addition only' subset (discussed in Section F.1) and its complementary 'not addition' subset. The experiments are performed under the same configuration as Section 6. Results for the addition subset and the complementary subset are presented in Table S7. In both subsets, our model outperforms the other models, indicating that although our dataset focuses on adding instructions, the inclusion of a large amount of high-quality editing data enhances performance for general editing tasks as well.

C. Multiple Object Addition

A straightforward extension of our model allows for adding multiple objects by applying it recurrently, each time with a different addition prompt. However, this approach poses

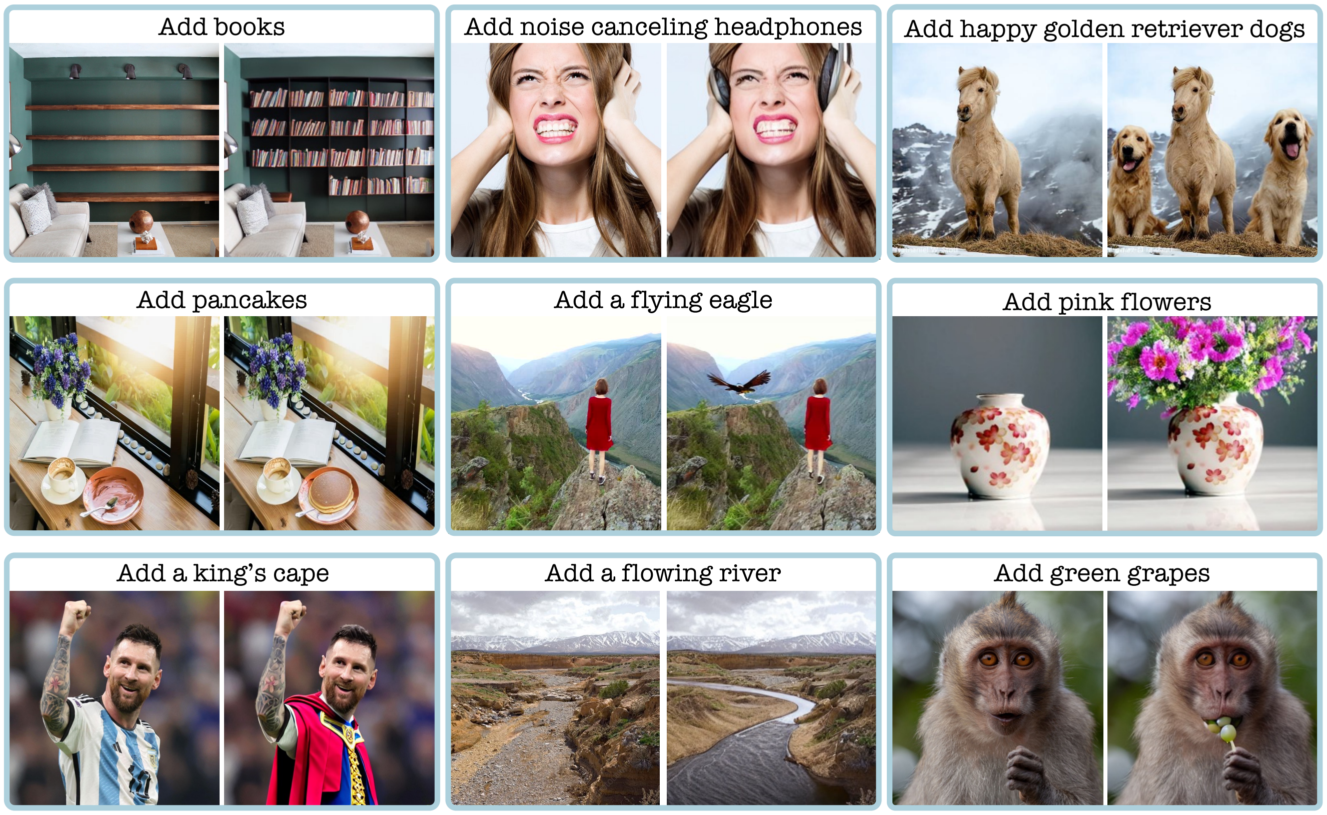


Figure S8. **Additional Object Addition Results of the Proposed Model.** The first two rows showcase outcomes from the model trained only with the PIPE dataset. The last row presents results from the same model after fine-tuning on the MagicBrush training set, as detailed in Section 5.2.

	Addition Subset					Non-Addition Subset				
Methods	L1$_\downarrow$	L2$_\downarrow$	CLIP-I$_\uparrow$	DINO$_\uparrow$	CLIP-T$_\uparrow$	L1$_\downarrow$	L2$_\downarrow$	CLIP-I$_\uparrow$	DINO$_\uparrow$	CLIP-T$_\uparrow$
IP2P	.100	.031	.860	.700	.289	.114	.038	.839	.742	.290
IP2P FT	.077	.028	.902	.867	.306	.083	.032	.895	.841	.300
Ours + IP2P FT	**.069**	**.024**	**.913**	**.889**	**.308**	**.075**	**.027**	**.905**	**.862**	**.303**

Table S7. **Global Editing Performance on Addition and Non-Addition MagicBrush Subsets.** Evaluation of our global editing model performance on both the add and complementary non-add instruction subsets of MagicBrush. The model, trained on the combined PIPE and IP2P datasets and fine-tuned on the MagicBrush training set, surpasses IP2P and the fine-tuned IP2P models in both subsets.

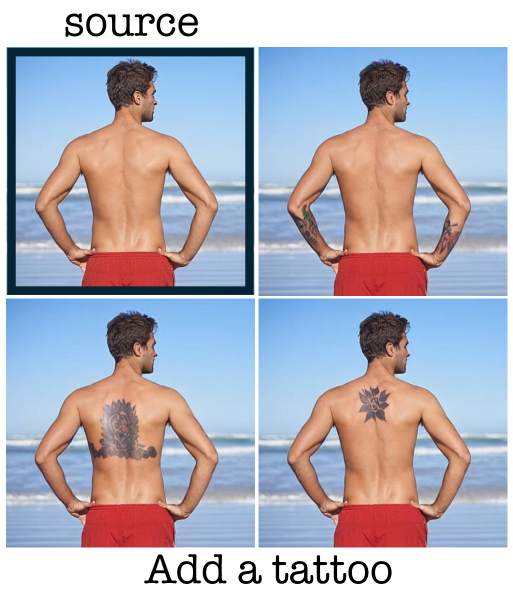


Figure S9. **Editing Diversity.** We generated three distinct edited images from the same source image, demonstrating the diversity of our model's outputs.

a challenge because each object addition requires a decode–encode cycle through the Stable Diffusion variational autoencoder (VAE), where each pass degrades image quality (see Fig. S15). To mitigate this, we perform all edits in latent space—encoding only before the first addition and decoding only after the final addition. Fig. S10 illustrates this process, demonstrating the successful addition of objects without intermediate decoding.

D. Limitations

Despite the impressive results produced by our model, several limitations remain. First, while our data curation pipeline improves robustness during the removal phase, it is not entirely error-free. Additionally, the model struggles with significant changes occurring far from the object but are affected by it. For instance, it handles nearby effects, like TV shadows (see Fig. S12), but struggles with larger shadows or distant reflections, as seen in the center images of Fig. S12. Similarly, object-object interactions are not

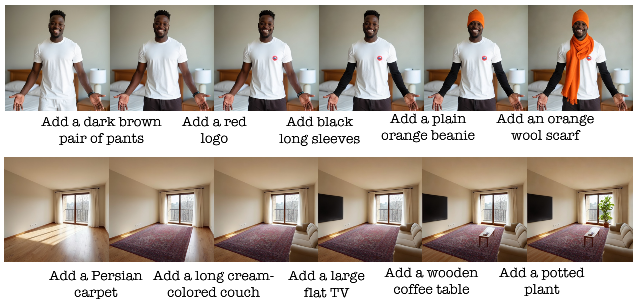


Figure S10. **Multiple objects.** Example of multiple object addition using latent editing, where successive objects are added without intermediate decoding and encoding.

always accurately handled (see the right images in the figure). These challenges stem from the dataset construction, as our method minimizes alterations outside the near-object region. Future work could explore inpainting both the object and distant regions influenced by it. We hope our work inspires future research to address these limitations.

E. PIPE Dataset

E.1. Creating Source-Target Image Pairs

We offer additional details on the post-removal steps described in Section 3.1. The post-removal process involves assessing the CLIP similarity between the class name of the removed object and the inpainted area. This assessment helps evaluate the quality of the object removal, ensuring no objects from the same class remain. To measure CLIP similarity for the inpainted area only, we counter the challenge of CLIP's unfamiliarity with masked images by reducing the background's influence on the analysis. We do this by adjusting the background to match the image's average color and integrating the masked area with this unified background color. A dilated mask smoothed with a Gaussian blur is employed to soften the edges, facilitating a more seamless and natural-looking blend.

To complement the CLIP score similarity, we introduce an additional measure that quantifies the shift in similarity before and after removal. Removals with a high pre-removal similarity score, followed by a comparatively lower

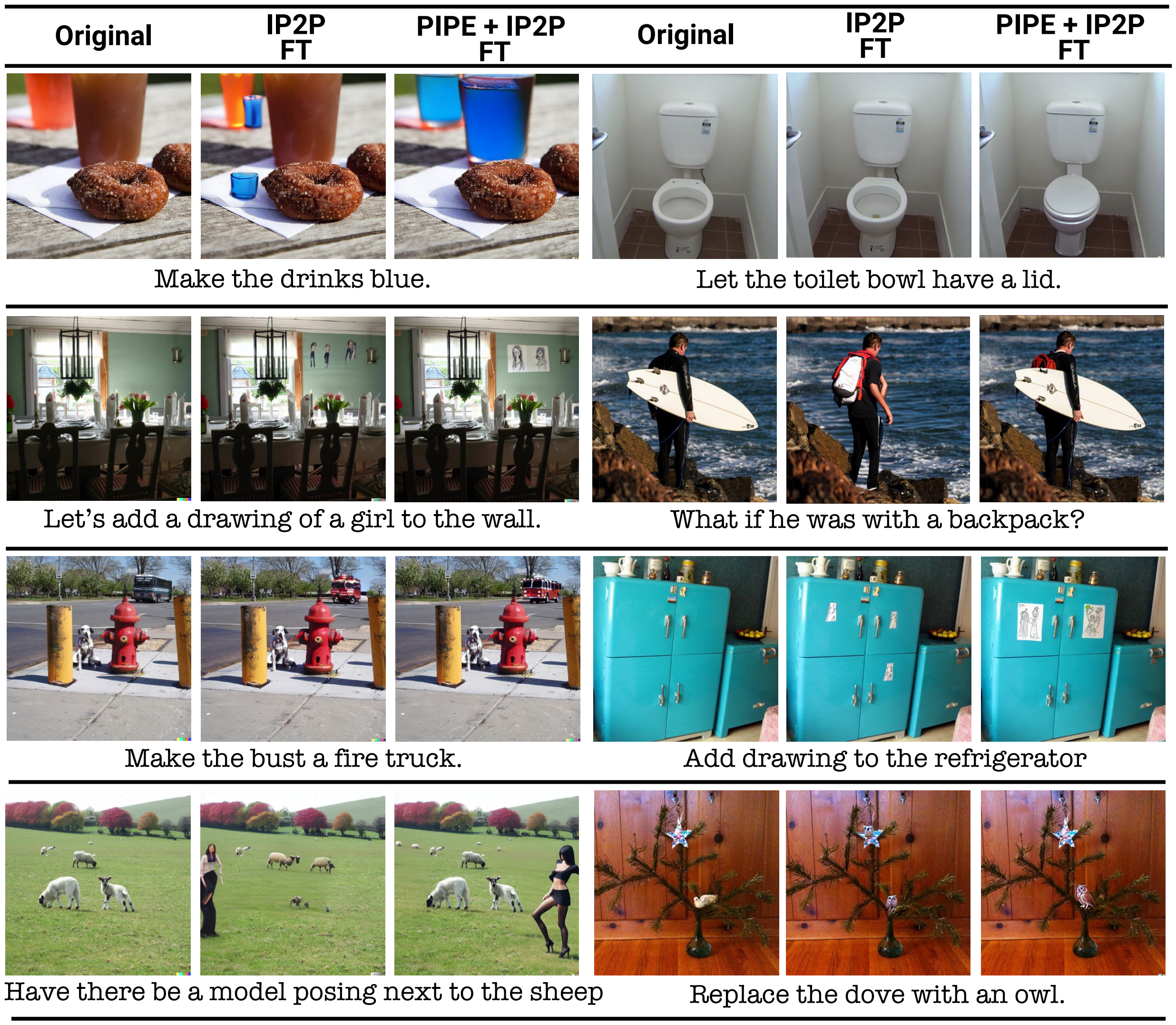

Original
IP2P FT
PIPE + IP2P FT
Original
IP2P FT
PIPE + IP2P FT
Make the drinks blue.
Let the toilet bowl have a lid.
Let's add a drawing of a girl to the wall.
What if he was with a backpack?
Make the bust a fire truck.
Add drawing to the refrigerator
Have there be a model posing next to the sheep
Replace the dove with an owl.

Figure S11. **Visual Comparison on General Editing Tasks.** The contribution of the PIPE dataset when combined with the IP2P dataset for general editing tasks, as evaluated on the full MagicBrush test set. The comparison is between a model trained on these merged datasets and a model trained solely on the IP2P dataset, with both models fine-tuned on the MagicBrush training set. The results demonstrate that, although the PIPE dataset focuses solely on object addition instructions, it enhances performance across a variety of editing tasks.

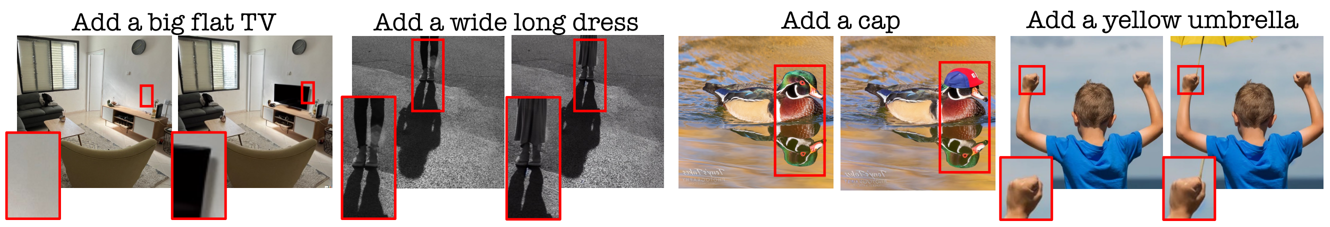


Figure S12. **Limitations.** Left: Successful shadow generation near the object. Center: Failures in generating shadows or reflections when distant from the object. Right: Failure in changing hand posture and maintaining the original one.

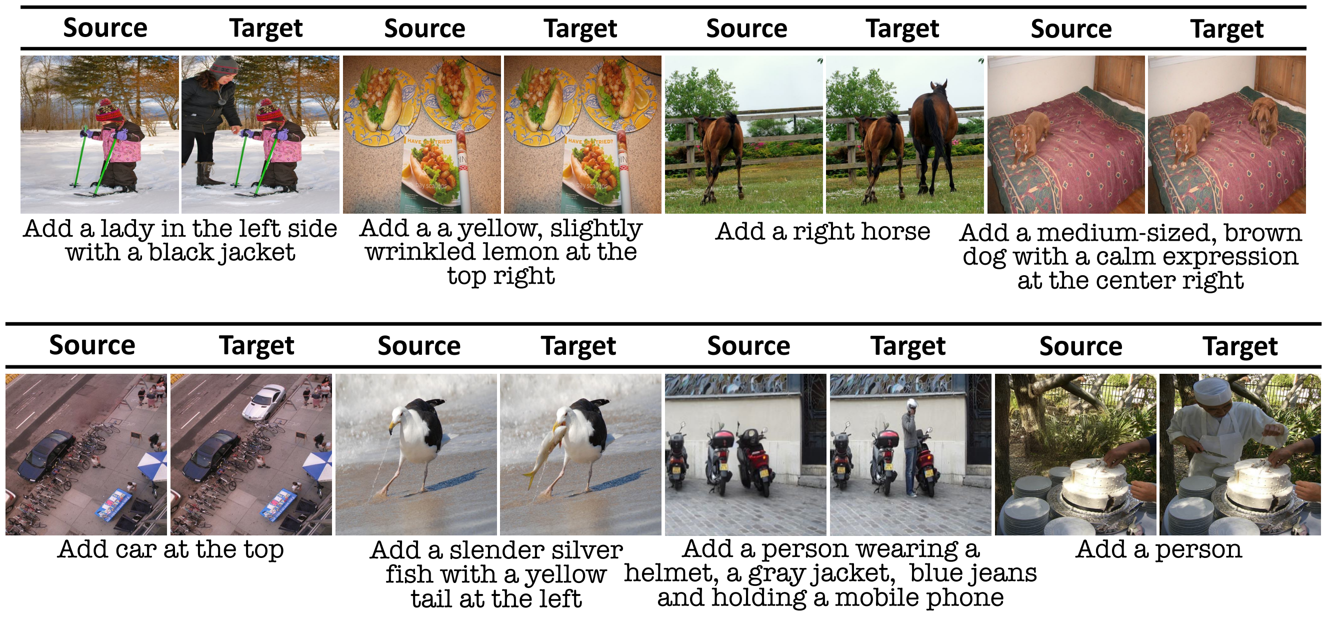

Source
Target
Source
Target
Source
Target
Source
Target
Add a lady in the left side with a black jacket
Add a a yellow, slightly wrinkled lemon at the top right
Add a right horse
Add a medium-sized, brown dog with a calm expression at the center right
Source
Target
Source
Target
Source
Target
Source
Target
Add car at the top
Add a slender silver fish with a yellow tail at the left
Add a person wearing a helmet, a gray jacket, blue jeans and holding a mobile phone
Add a person

Figure S13. **Additional PIPE Datasets Examples.**

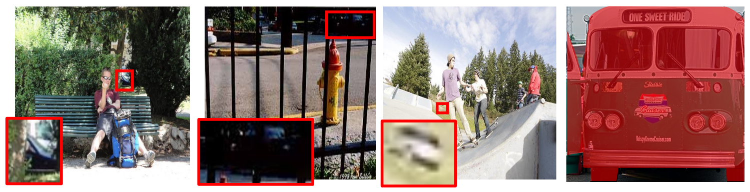


Figure S14. **Pre-Removal Filtered Examples.** Left: Objects with non-informative view and low CLIP Object similarity. Right: Extremely small and large objects, unsuitable for our dataset.

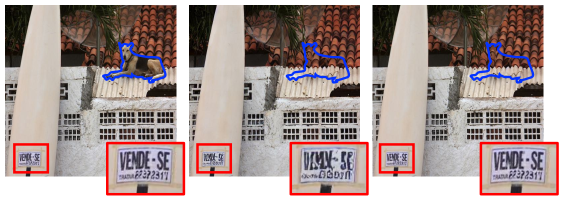


Figure S15. **Consistency Enforcement Examples.** From left to right: original image, inpainted dog image, inpainted image after alpha blending.

yet significant post-removal score are not filtered, even though they exceed the threshold. This method allows for the efficient exclusion of removals, even when other objects of the same class are in close spatial proximity.

Figure S21 and Figure S22 present the figures discussed in Section 3.1 related to filtering thresholds and their justification. Table S8 reports the number of images before and after each filtering stage.

Table S8. **Statistics on the dataset before and after each Filtering step.**

Initial	Pre-Removal	Consensus	MM CLIP	Importance
4,646K	1,494K	1,101K	986K	888K

E.2. VLM-LLM Based Instructions

Using a VLM and an LLM, we convert the class names of objects from the segmentation dataset into detailed natural language instructions (Section 3.2). Initially, for each image, we present the masked image (featuring only the object) to CogVLM with the prompt: "`Accurately describe the main characteristics of the <class name>. Use few words which best describe the <class- name>`". This process yields an in-depth description centered on the object, highlighting key attributes such as shape, color, and texture. Subsequently, this description is provided to the LLM along with human-crafted prompts for In-Context Learning (ICL), to generate succinct and clear instructions. The implementation of the ICL mechanism is detailed in Table S9.

Furthermore, we enrich the instructions by including a coarse language-based description of the object's location within the image, derived from the given mask. To accomplish this, we split the image into a nine-section grid and assign each section a descriptive label (e.g., top-right). This spatial description is then randomly appended to the instruction with a 25% probability during the training process.

E.3. Integrating Instruction Types

As detailed in Section 3.2, we construct our instructions using three approaches: (i) class name-based (ii) VLM-LLM based, and (iii) manual reference-based. These three cate-

Table S9. **In-Context Learning Prompt**. (Top) We provide the model with five examples of captions and their corresponding human-annotated responses. (Bottom) We introduce it with a new caption and request it to provide an instruction.

[**USER**]: Convert the following sentence into a short image addition instruction:
¡caption 0¿.
Use straightforward language and describe only the ¡class name 0¿.
Ignore surroundings and background and avoid pictorial description.
[**ASSISTANT**]: ¡example response 0¿

⋮

[**USER**]: Convert the following sentence into a short image addition instruction:
¡caption 4¿.
Use straightforward language and describe only the ¡class name 4¿.
Ignore surroundings and background and avoid pictorial description.
[**ASSISTANT**]: ¡example response 4¿

- -

[**USER**]: Convert the following sentence into a short image addition instruction:
¡new caption¿.
Use straightforward language and describe only the ¡new class name¿.
Ignore surroundings and background and avoid pictorial description.
[**ASSISTANT**]:

gories are then integrated to assemble the final dataset. The dataset includes 887,773 instances each from Class name-based and VLM-LLM-based methods, with an additional 104,373 from Manual reference-based instructions.

E.4. Additional Examples

In Figure S13, we provide further instances of the PIPE dataset that complement those in Figure 5.

F. Implementation Details

As noted in Section 4, the training of our editing model is initialized with the SD v1.5 model. Conditions are set with $c_T = \varnothing$, $c_I = \varnothing$, and both $c_T = c_I = \varnothing$ occurring with a 5% probability each. The input resolution during training is adjusted to 256, applying random cropping for variation. Each GPU manages a batch size of 128. The model undergoes training for 60 epochs, utilizing the ADAM optimizer. It employs a learning rate of $5 \cdot 10^{-5}$, without a warm-up phase. Gradient accumulation is set to occur over four steps preceding each update, and the maximum gradient norm is clipped at 1. Utilizing eight NVIDIA A100 GPUs, the total effective batch size, considering the per-GPU batch size, the number of GPUs, and gradient accumulation steps, reaches 4096 $(128 \cdot 8 \cdot 4)$.

For the fine-tuning phase on the MagicBrush training set (Section 5.2), we adjust the learning rate to 10^{-6} and set the batch size to 8 per GPU, omitting gradient accumulation, and train for 250 epochs.

F.1. MagicBrush Subset

To initially focus our analysis on the specific task of object addition, we applied an automated filtering process to the MagicBrush dataset. This process aims to isolate image pairs and associated instructions that exclusively pertained to object addition. To ensure an unbiased methodology, we applied an automatic filtering rule across the entire dataset. The filtering criterion applied retained instructions explicitly containing the verbs ”add” or ”put,” indicating object addition. Concurrently, instructions with ”remove” were excluded to avoid object replacement scenarios, and those with the conjunction ”and” were omitted to prevent cases involving multiple instructions.

F.2. Evaluation

In our comparative analysis in Section 5.2, we assess our model against leading instruction-following image editing models. To ensure a fair and consistent evaluation across all models, we employed a fixed seed (0) for all comparisons.

Our primary analysis focuses on two instruction-guided models, IP2P [4] and Hive [73]. For IP2P, we utilized the Hugging Face diffusers model and pipeline[3], adhering to the default inference parameters. Similarly, for Hive, we employed the official implementation provided by the authors[4], with the documented default parameters.

Our comparison extends to models that utilize global descriptions: VQGAN-CLIP [9] Null-Text-Inversion [41], Pix2PixZero [46], Edit-Freindly DDPM [22] and SDEdit [40]. These models were chosen for evaluation within the MagicBrush dataset, as global descriptions are not available in both the OPA and our PIPE dataset. For

[3]https://hf.co/docs/diffusers/training/instructpix2pix
[4]https://github.com/salesforce/HIVE

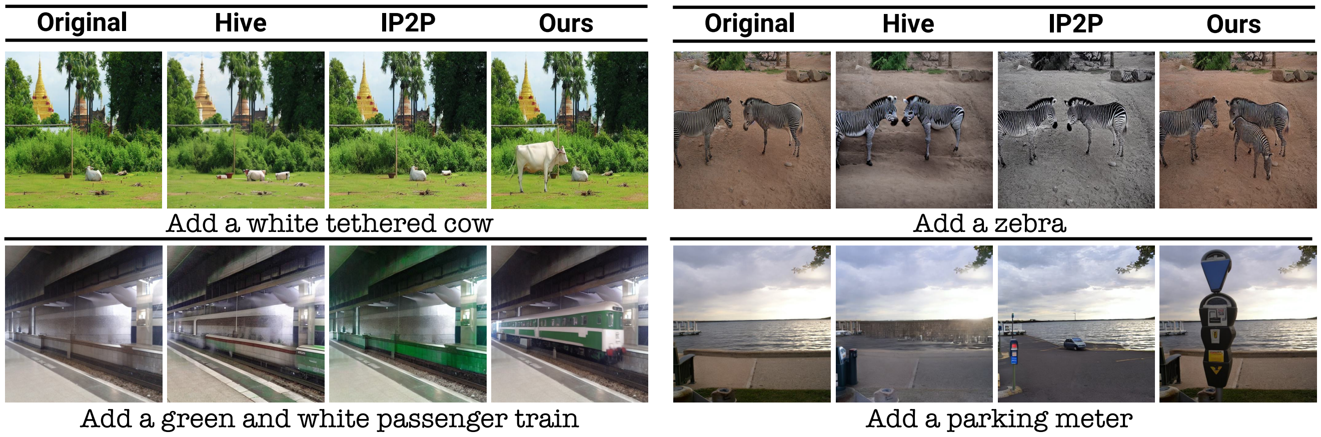


Figure S16. **Visual Comparison of the Proposed Model on PIPE Test Set.** The visual evaluation highlights the effectiveness of our method against other leading models on the PIPE test set. Our model excels in adhering closely to specified instructions and accurately generating objects in terms such as style, scale, and location.

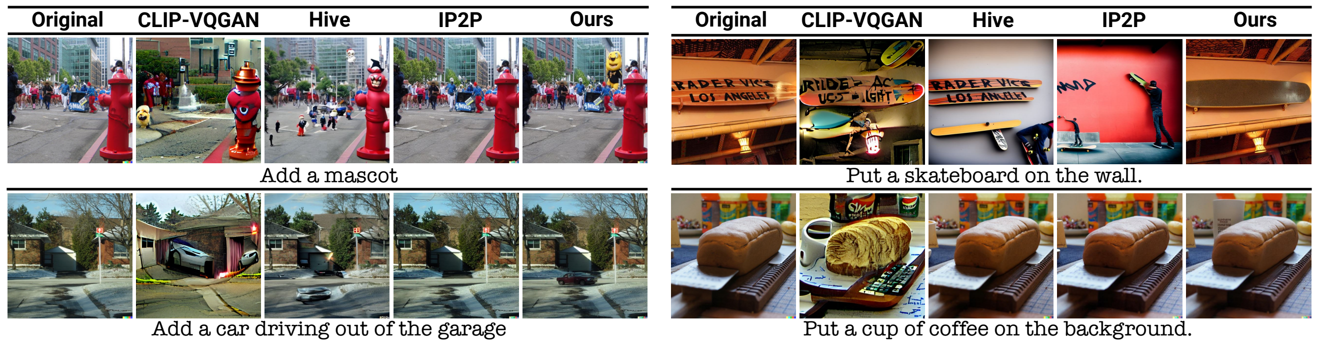

Original
CLIP-VQGAN
Hive
IP2P
Ours
Add a mascot
Put a skateboard on the wall.
Add a car driving out of the garage
Put a cup of coffee on the background.

Figure S17. **Visual Comparison of the Proposed Model on MagicBrush Test Subset.** Our method versus leading models within the MagicBrush object addition test subset. It illustrates our model's superior generalization across varied instructions and datasets, outperforming the other approaches.

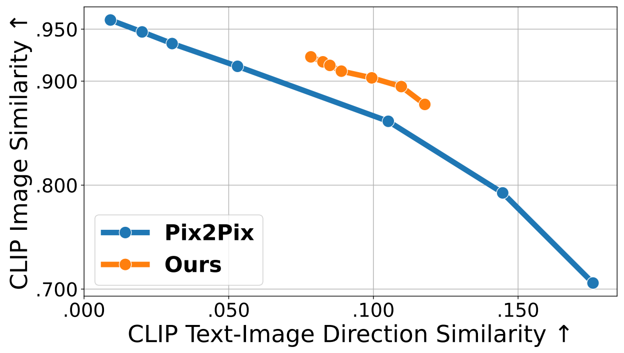


Figure S18. **Model Consistency-Instruction Trade-off:** Trade-off between consistency with the input image (Y-axis) and edit adherence (X-axis) for IP2P and our model on the MagicBrush test subset. Text guidance is fixed at 7, and image guidance ranges from 1 to 2.5.

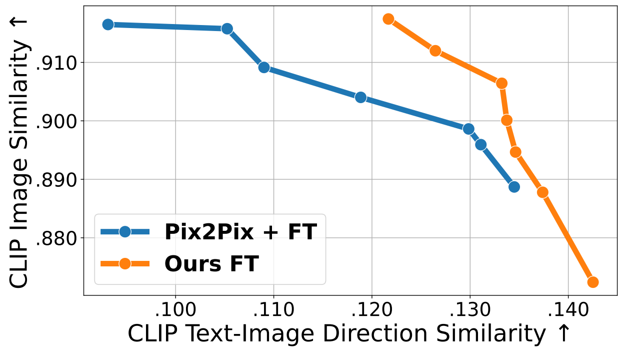


Figure S19. **Finetuned-Model Consistency-Instruction Trade-off:** Trade-off between consistency with the input image (Y-axis) and edit adherence (X-axis) for IP2P and our model, both fine-tuned on the MagicBrush training set and tested on its test subset. Text guidance is fixed at 7, and image guidance ranges from 1 to 2.5.

Model	VQGAN-CLIP	SDEdit	NTI	P2P-Z	EFD	Hive	IP2P	Ours
VQAScore	0.7675	0.6114	0.6008	0.5356	0.6792	0.5822	0.5408	0.7045

Table S10. **VQAScore Metric.** We use VQAScore [34] as a VQA-based alignment metric to further evaluate our method.

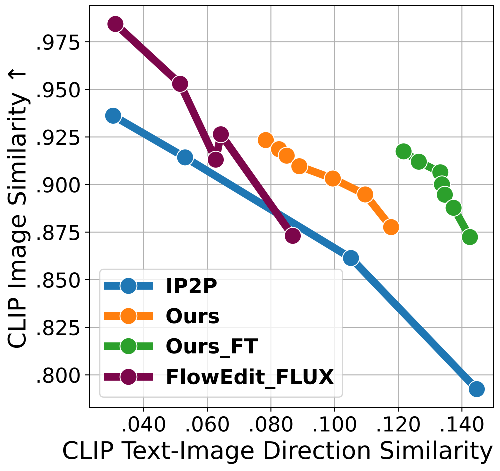


Figure S20. **Comparison to FLUX-Based Method.** We compared our model to a FLUX-based approach, specifically the recently released FlowEdit [27]. As shown in the plot, our model significantly outperforms FlowEdit.

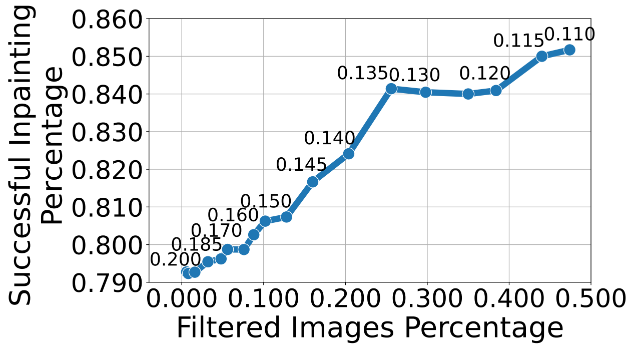


Figure S21. **Concensus Filtering Success for varying Thresholds**

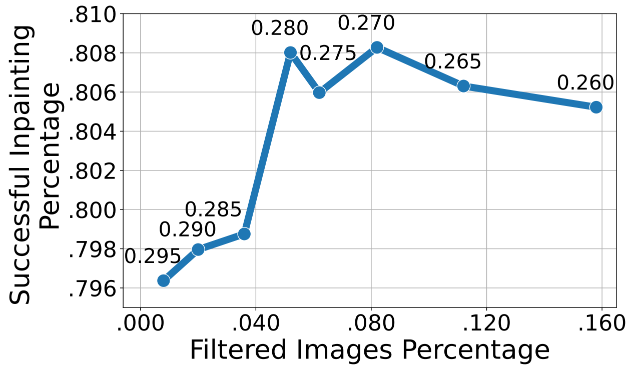


Figure S22. **Multimodal CLIP Filtering Success for varying Thresholds**

VQGAN-CLIP[5], Null-Text-Inversion[6] and Edit-Freindly DDPM[7], we used the official code base with the default hyperparameters. For SDEdit[8] and Pix2PixZero[9], we used the image-to-image pipeline of the Diffusers library with the default parameters.

We also evaluated our fine-tuned model against the MagicBrush fine-tuned model, as documented in [72]. Although this model does not serve as a measure of generalizability, it provides a valuable benchmark within the specific context of the MagicBrush dataset. For this comparison, we employed the model checkpoint and parameters as recommended on the official GitHub repository of the MagicBrush project[10]. In Figure S16 and Figure S17, we provide additional qualitative examples on the tested datasets to complement the ones in Figure 3. We further assess the model's performance on the MagicBrush subset using the same CLIP Image similarity versus Directional CLIP similarity measure, as explained in Section 6. We plot this measure to compare the IP2P model with our model in Figure S18 and the MagicBrush fine-tuned models in Figure S19. As shown in both comparisons, our models present a better trade-off between consistency with the input image and adherence to the edit instruction, achieving higher consistency with the instruction for the same similarity to the input image.

To further evaluate our method with more advanced metrics, we extend Table 2 and report the VQAScore [34], as shown in Table S10. Under this metric, our approach maintains its favorable performance, except for VQGAN-CLIP, which, as discussed in Section 5.2, tends to deviate substantially from the original image. Furthermore, we extend Figure 6 by adding a comparison between our model and a FLUX-based [30] approach, the recently released FlowEdit [27], demonstrating superior performance.

G. Instructions Ablation

We examine the impact of employing our VLM-LLM pipeline, detailed in Section 3.2, for generating natural language instructions. The outcomes of the pipeline, termed "long instructions", are compared with brief, class name-based instructions (*e.g.*, "`Add a cat`"), referred to as "short instructions". In Table S11, we assess a model

[5]https://github.com/nerdyrodent/VQGAN-CLIP
[6]https://github.com/google/prompt-to-prompt/blob/main/null_text_w_ptp.ipynb
[7]https://github.com/inbarhub/DDPM_inversion
[8]https://hf.co/docs/diffusers/en/api/pipelines/stable_diffusion/img2img
[9]https://hf.co/docs/diffusers/main/en/api/pipelines/pix2pix_zero
[10]https://github.com/OSU-NLP-Group/MagicBrush

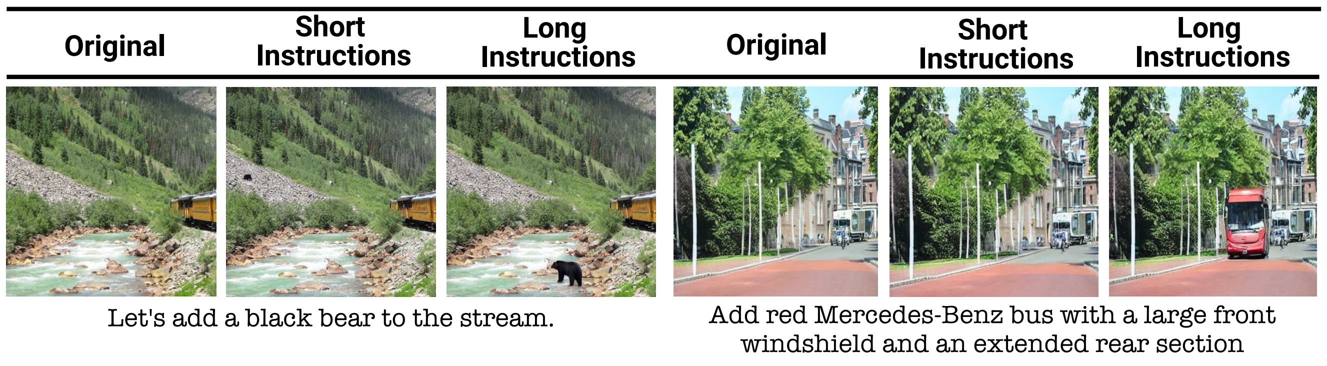

Original
Short Instructions
Long Instructions
Original
Short Instructions
Long Instructions
Let's add a black bear to the stream.
Add red Mercedes-Benz bus with a large front windshield and an extended rear section

Figure S23. **Instructions Ablation Examples.** Qualitative comparison of model performance when trained on 'short' template-based instructions versus 'long' instructions generated through our VLM-LLM pipeline. Models trained on the latter exhibit superior performance in interpreting complex instructions and closely aligning object additions with editing requests.

Train Instructions Type	L1 ↓	L2 ↓	CLIP-I ↑	DINO↑	CLIP-T ↑
Short Instructions	0.083	0.028	**0.900**	**0.856**	0.300
Long Instructions	**0.072**	**0.025**	**0.900**	0.852	**0.302**

Table S11. **Instructions Ablation Analysis.** A quantitative comparative analysis of model performance, comparing training on 'short' class-based instructions to 'long' instructions generated using the VLM and LLM pipeline. This analysis was performed on MagicBrush subset. The results demonstrate that training with VLM-LLM-based instructions significantly enhances performance, thereby confirming its effectiveness.

trained on the PIPE image pairs, comparing its performance when trained with either long or short inputs. The models are evaluated on MagicBrush subset. As expected, training with long instructions leads to improved performance on MagicBrush. This demonstrates that training with comprehensive instructions generated by our VLM-LLM mechanism benefits at inference time. In addition to quantitative results, we provide qualitative results of both models in Figure S23. As illustrated, the model trained with long instructions shows superior performance in interpreting complex instructions that include detailed descriptions and location references, such as "Let's add a black bear to the stream".

H. Human Evaluation

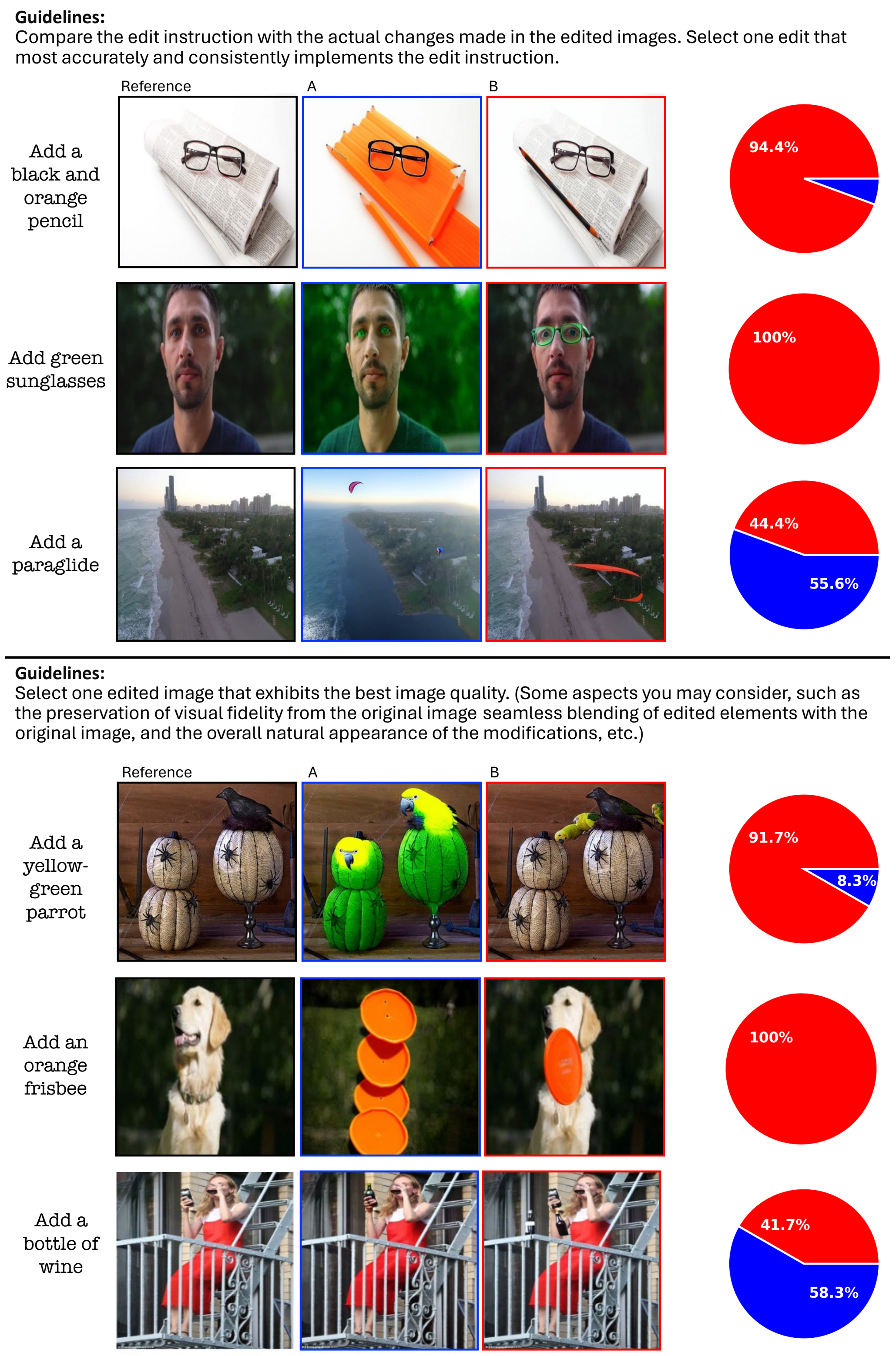
While quantitative metrics are important for evaluating image editing performance, they do not fully capture human satisfaction with the edited outcomes. To this end, we conduct a human evaluation survey, as explained in Section 5.4, comparing our model with IP2P and hive (Tab. S12). Following [72], we pose two questions: one regarding the execution of the requested edit and another concerning the overall quality of the resulting images. Figure S24 illustrates examples from our human survey along with the questions posed. Overall, our method leads to better results for human perception. Interestingly, as expected due to how PIPE was constructed, our model maintains a higher level of consistency with the original images in both its success and failure cases. For example, in the third row of Figure S24, while IP2P generates a more reliable paraglide, it fails to preserve the original background.

Methods	**Edit faithfulness**		**Quality**	
	Overall [%]	Per image	Overall [%]	Per-image
Hive	25.9	21	24.8	22
Ours	**74.1**	**79**	**75.2**	**78**

Table S12. **Human Evaluation against Hive.**

I. Social Impact and Ethical Consideration

Using PIPE or the model trained with it significantly enhances the ability to add objects to images based on textual instructions. This offers considerable benefits, enabling users to seamlessly and quickly incorporate objects into images, thereby eliminating the need for specialized skills or expensive tools. The field of image editing, specifically the addition of objects, presents potential risks. It could be exploited by malicious individuals to create deceptive or harmful imagery, thus facilitating misinformation or adverse effects. Users are, therefore, encouraged to use our findings responsibly and ethically, ensuring that their applications are secure and constructive. Furthermore, PIPE, was developed using a VLM [65] and an LLM [24], with the model training starting from a SD checkpoint [52]. Given that the models were trained on potentially biased or explicit, unfiltered data, the resulting dataset may reflect these original biases.



Guidelines:
Compare the edit instruction with the actual changes made in the edited images. Select one edit that most accurately and consistently implements the edit instruction.

Guidelines:
Select one edited image that exhibits the best image quality. (Some aspects you may consider, such as the preservation of visual fidelity from the original image seamless blending of edited elements with the original image, and the overall natural appearance of the modifications, etc.)

Figure S24. **Human Evaluation Examples**. Examples of the qualitative survey against IP2P alongside the response distribution (our method in red and the baseline in blue). The examples include both successful and failed cases of our model. The first three top examples correspond with a question focused on the edit completion, and the three bottom ones on the resulting image quality.