OMNIEDIT: BUILDING IMAGE EDITING GENERALIST MODELS THROUGH SPECIALIST SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Instruction-guided image editing methods have demonstrated significant potential by training diffusion models on automatically synthesized or manually annotated image editing pairs. However, these methods remain far from practical, real-life applications. We identify three primary challenges contributing to this gap. Firstly, existing models have limited editing skills due to the biased synthesis process. Secondly, these methods are trained with datasets with a high volume of noise and artifacts. This is due to the application of simple filtering methods like CLIP-score. Thirdly, all these datasets are restricted to a single low resolution and fixed aspect ratio, limiting the versatility to handle real-world use cases. In this paper, we present OMNI-EDIT, which is an omnipotent editor to handle seven different image editing tasks with any aspect ratio seamlessly. Our contribution is in four folds: (1) OMNI-EDIT is trained by utilizing the supervision from seven different specialist models to ensure task coverage. (2) we utilize importance sampling based on the scores provided by large multimodal models (like GPT-4o) instead of CLIP-score to improve the data quality. (3) we propose a new editing architecture called EditNet to greatly boost the editing success rate, (4) we provide images with different aspect ratios to ensure that our model can handle any image in the wild. We have curated a test set containing images of different aspect ratios, accompanied by diverse instructions to cover different tasks. Both automatic evaluation and human evaluations demonstrate that OMNI-EDIT can significantly outperform all the existing models.

1 INTRODUCTION

Image editing, particularly when following user instructions to apply semantic transformations to real-world photos, has seen significant advancements. Recently, text-guided image editing (Brooks et al., 2023) has gained prominence over traditional methods such as mask-based or region-based editing (Meng et al., 2022). With the rise of diffusion models (Rombach et al., 2022; Podell et al., 2024; Chen et al., 2024a; Sauer et al., 2024), numerous diffusion-based image editing techniques have emerged. Generally, they can be roughly divided into two types: (1) Inversion-based methods (Parmar et al., 2023; Kawar et al., 2023; Gal et al., 2023; Xu et al., 2023; Tumanyan et al., 2023; Tsaban & Passos, 2023) propose to perform zero-shot image editing by inverting the diffusion process and manipulating the attention map in the intermediate diffusion steps to achieve desired editing goal. (2) End-to-end methods (Brooks et al., 2023; Zhang et al., 2024a; Sheynin et al., 2024; Zhao et al., 2024; Fu et al., 2024) propose to fine-tune an existing diffusion model on large-scale image editing pairs to learn the editing operation in an end-to-end fashion. End-to-end methods have generally achieved better performance than inversion-based methods and gained higher popularity.

Despite their effectiveness, end-to-end methods face a significant limitation: the scarcity of human-annotated image editing pairs. As a result, all current end-to-end approaches depend on synthetic training data. For instance, existing datasets are synthesized using techniques such as Prompt2Prompt (Hertz et al., 2023) or mask-based editing models like SD-Inpaint (Rombach et al., 2022), and DALLE-2/3 (Ramesh et al., 2022; Betker et al., 2023). However, these synthetic data generation pipelines exhibit significant biases, resulting in the following limitations:

Limited Editing Capabilities: The synthetic data is heavily influenced by the underlying generation models. For example, Prompt2Prompt struggles with localized edits, such as adding, removing, or

Table 1: Comparison of OMNI-EDIT with all the existing end-to-end image editing models.

Property	InstructP2P	MagicBrush	UltraEdit	MGIE	HQEdit	CosXL	OMNI-EDIT
Training Dataset Properties							
Real Image?	✗	✓	✓	✓	✗	✗	✓
Any Res?	✗	✗	✗	✗	✗	✗	✓
High Res?	✗	✗	✗	✗	✓	✗	✓
Fine-grained Image Editing Skills							
Obj-Swap	★★☆	★★☆	★★☆	★½☆	★★☆	★☆☆	★★½
Obj-Add	★☆☆	★★☆	★☆☆	★½☆	★☆☆	★☆☆	★★½
Obj-Remove	★☆☆	★★☆	★☆☆	★½☆	★☆☆	★☆☆	★★½
Attribute	★★☆	★☆☆	★★☆	★½☆	★☆☆	★☆☆	★★½
Back-Swap	★★☆	★★☆	★★☆	★½☆	★★☆	★★☆	★★½
Environment	★☆☆	★☆☆	★☆☆	★½☆	★☆☆	★★☆	★★½
Style	★★☆	★☆☆	★★☆	★½☆	★☆☆	★★½	★★½

swapping objects, while SD-Inpaint and DALLE-2 are ineffective at global edits, such as style or background changes. As a result, models trained on such data inherit these limitations.

Poor Data Quality Control: Most approaches use simplified filtering mechanisms like CLIP-score (Radford et al., 2021) or DINO-score (Caron et al., 2021) to automatically select training samples. However, recent studies (Ku et al., 2024) show that these metrics exhibit poor correlation with actual data quality, leading to suboptimal training data that negatively impacts the model.

Lack of Support for Varying Resolutions: All current models are trained on square image editing pairs, making their generalization to non-square images poor.

In our preliminary studies, we curate a few prompts for seven different desired tasks to observe their success rate across the board. We show our findings in Table 1. This show that these models are truly biased in their skills caused by the underlying synthesis pipeline.

In this paper, we introduce OMNI-EDIT, a novel model designed to address these challenges through four key innovations:

1. Specialist-to-Generalist Supervision: We propose learning a generalist editing model, OMNI-EDIT, by leveraging supervision from multiple specialist models. Unlike previous approaches that rely on a single expert, we conduct an extensive survey and construct (or train) seven experts, each specializing in a different editing task. These specialists provide supervisory signals to OMNI-EDIT.

2. Importance Sampling: To ensure high-quality training data, we employ large multimodal models to assign quality scores to synthesized samples. Given the computational cost of GPT-4o (Achiam et al., 2023), we first distill its scoring ability into InternVL2 (Chen et al., 2024b) through medium-sized samples. Then we use the InternVL2 model for large-scale scoring.

3. EditNet Architecture: We introduce EditNet, a novel diffusion-transformer-based architecture (Peebles & Xie, 2022) that facilitates interaction between the control branch and the original branch via intermediate representations. This architecture enhances OMNI-EDIT 's ability to comprehend diverse editing tasks.

4. Support for Any Aspect Ratio: During training, we incorporate a mix of images with varying aspect ratios as well as high resolution, ensuring that OMNI-EDIT can handle images of any aspect ratio with any degradation in the output quality.

We curate an image editing benchmark OMNI-EDIT-BENCH, which contains diverse images of different resolutions and diverse prompts that cover all the listed editing skills. We perform comprehensive automatic and human evaluation to show the significant boost of OMNI-EDIT over the existing baseline models like CosXL-Edit (Boesel & Rombach, 2024), UltraEdit (Zhao et al., 2024), etc.

2 PRELIMINARIES

2.1 TEXT-TO-IMAGE DIFFUSION MODELS

Diffusion models (Song et al., 2021; Ho et al., 2020) are a class of latent variable models parameterized by θ, defined as $p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T}$, where $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ represents the original data, and $\mathbf{x}_1, \ldots, \mathbf{x}_T$ are progressively noisier latent representations of the input image $\mathbf{x}_0$. Through-

Figure 1: **Editing high-resolution multi-aspect images with OMNI-EDIT** OMNI-EDIT is an instruction-based image editing generalist capable of performing diverse editing tasks across different aspect ratios and resolutions. It accurately follows instructions while preserving the original image's fidelity. We suggest zooming in for better visualization.

out the process, the dimensionality of $\mathbf{x}_0$ and the latent variables $\mathbf{x}_{1:T}$ remains consistent, with $\mathbf{x}_{0:T} \in \mathbb{R}^d$, where d corresponds to the product of the image's height, width, and channels. The forward (diffusion) process, denoted as $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$, is a predefined Markov chain that incrementally adds Gaussian noise to the data according to a pre-defined schedule $\{\beta_t\}_{t=1}^T$. The process of forward diffusion is defined as:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t \mathbf{I}), \tag{1}$$

where $\mathcal{N}$ denotes a Gaussian distribution, and β_t controls the amount of noise added at each step. The objective of diffusion models is to reverse this diffusion process by learning the distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, which enables the reconstruction of the original data $\mathbf{x}_0$ from a noisy latent $\mathbf{x}_t$. This reduces to a denoising problem where the model ϵ_θ is trained to denoise the sample $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_0)$ back into $\mathbf{x}_0$. The maximum log-likelihood training objective breaks down to minimizing the weighted mean squared error between the model's prediction $\hat{\mathbf{x}}_\theta(\mathbf{x}_t, c)$ and the true data $\mathbf{x}_0$:

$$\arg\max_\theta \log p_\theta(\mathbf{x_0}|c) = \arg\min_\theta \mathbb{E}_{(\mathbf{x}_0,c)\sim\mathcal{D}} \left[\mathbb{E}_{\epsilon,t}\left[w_t \cdot \|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, c) - \mathbf{x}_0\|_2^2\right]\right], \tag{2}$$

where $(\mathbf{x}_0, c)$ pairs come from the dataset $\mathcal{D}$, with c representing the text prompt. The term w_t is a weighting factor applied to the loss at each timestep t. For simplicity, prior papers (Song et al., 2021; Ho et al., 2020; Karras et al., 2022) will set w_t to be 1.

2.2 Instruction-Based Image Editing in Supervised Learning

Instruction-based image editing can be formulated as a supervised learning problem. Existing methods (Brooks et al., 2023; Zhang et al., 2024a) often adopt a paired training dataset of text editing instructions and images before and after the edit. An image editing diffusion model is then trained on this dataset. The latent diffusion objective is defined as:

$$\arg\max_{\theta} \log p_\theta(\mathbf{x}_0'|\mathbf{x}_0, c) = \arg\min_{\theta} \mathbb{E}_{(\mathbf{x}_0', \mathbf{x}_0, c)\sim\mathcal{D}} \left[\mathbb{E}_{\epsilon,t}\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, c) - \mathbf{x}_0'\|_2^2\right], \tag{3}$$

where $(\mathbf{x}_0', \mathbf{x}_0, c)$ triples are sampled from the dataset $\mathcal{D}$ with $\mathbf{x}_0$ denoting the source image, c denoting the editing instruction and $\mathbf{x}_0'$ denoting the target image.

3 Learning with Specialist Supervision

In this section, we introduce the entire specialist-to-generalist learning framework to build Omni-Edit. We describe the overall learning objective in subsection 3.1. We then describe how we learn the specialists in subsection 3.2 and the importance weighting function in subsection 3.3. In Figure 2, we show the overview of the Omni-Edit training pipeline.

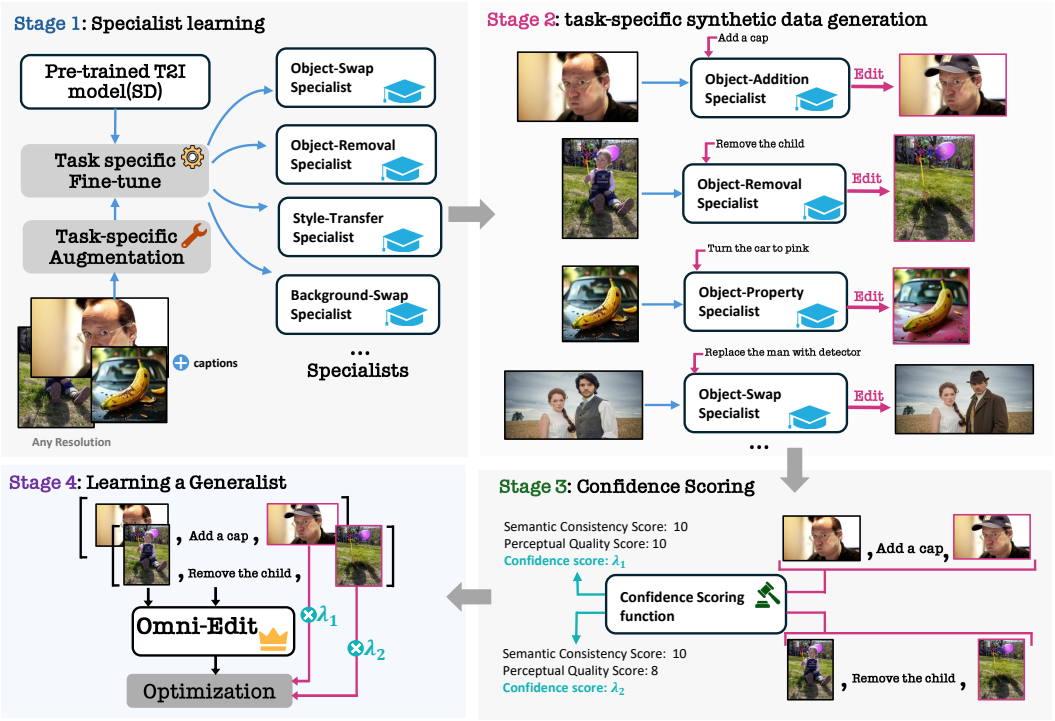

Figure 2: Overview of the Omni-Edit training pipeline.

3.1 Learning Objective

We assume there is a groundtruth editing model $p(\mathbf{x}'|\mathbf{x}, c)$, which can perform any type of editing tasks perfectly according to the instruction c. Our goal is to minimize the divergence between $p_\theta(\mathbf{x}'|\mathbf{x}, c)$ with $p(\mathbf{x}'|\mathbf{x}, c)$ by updating the parameters θ:

$$L(\theta) := \sum_{\mathbf{x},c} D_{KL}(p(\mathbf{x}'|\mathbf{x}, c)\|p_\theta(\mathbf{x}'|\mathbf{x}, c)) = -\sum_{\mathbf{x},c}\sum_{\mathbf{x}'} p(\mathbf{x}'|\mathbf{x}, c)\log p_\theta(\mathbf{x}'|\mathbf{x}, c) + C \tag{4}$$

Table 2: Task Definitions and Examples

Editing Tasks	Definition	Instruction c Example
Object Swap	c describes an object to replace by specifying both the object to remove and the new object to add, along with their properties such as appearance and location.	Replace the black cat with a brown dog in the image.
Object Removal	c describes which object to remove by specifying the object's properties such as appearance, location, and size.	Remove the black cat from the image.
Object Addition	c describes a new object to add by specifying the object's properties such as appearance and location.	Add a red car to the left side of the image.
Attribute Modification	c describes how to modify the properties of an object, such as changing its color, facial expression, material or texture.	Change the blue car to a red car.
Background Swap	c describes how to replace the background of the image, specifying what the new background should be.	Replace the background with a space-ship interior.
Environment Change	c describes a change to the overall environment, such as the weather, lighting, or season, without altering specific objects.	Change the scene from daytime to nighttime.
Style Transfer	c describes how to apply a specific artistic style or visual effect to the image, altering its overall appearance while keeping the content the same.	Apply a watercolor painting style to the image.

where C is a constant, which we leave out in the following derivation. However, since we don't have access to $p(\mathbf{x}'|\mathbf{x}, c)$, we adopt importance sampling for approximation:

$$\begin{aligned} L(\theta) &= -\sum_{\mathbf{x},c}\sum_{\mathbf{x}'} q(\mathbf{x}'|\mathbf{x}, c)\frac{p(\mathbf{x}'|\mathbf{x}, c)}{q(\mathbf{x}'|\mathbf{x}, c)}\log p_\theta(\mathbf{x}'|\mathbf{x}, c) \\ &\approx -\mathbb{E}_{(\mathbf{x},c)\sim D}\left[\mathbb{E}_{\mathbf{x}'\sim q(\mathbf{x}'|\mathbf{x},c)}\left[\lambda(\mathbf{x}', \mathbf{x}, c)\log p_\theta(\mathbf{x}'|\mathbf{x}, c)\right]\right] \\ &\approx -\mathbb{E}_{(\mathbf{x},c)\sim D}\left[\mathbb{E}_{\mathbf{x}'\sim q_s(\mathbf{x}'|\mathbf{x},c)}\left[\lambda(\mathbf{x}', \mathbf{x}, c)\log p_\theta(\mathbf{x}'|\mathbf{x}, c)\right]\right] \end{aligned} \tag{5}$$

where $q(\mathbf{x}'|\mathbf{x}, c)$ is the proposal distribution and $\lambda(\cdot)$ is the importance function. To better approximate the groundtruth distribution $p(\mathbf{x}'|\mathbf{x}, c)$, we propose to use an ensemble model $q(\mathbf{x}'|\mathbf{x}, c)$. In essence, $q(\mathbf{x}'|\mathbf{x}, c) := q_s(\mathbf{x}'|\mathbf{x}, c)$, where q_s is a specialist distribution decided by the type of the instruction c (e.g. object removal, object addition, stylization, etc). Combing with Equation 3, our objective can be rewritten as:

$$\arg\min_\theta L(\theta) = \arg\min_\theta \mathbb{E}_{(\mathbf{x},c)\sim D}\mathbb{E}_{\mathbf{x}'\sim q_s(\mathbf{x}'|\mathbf{x},c)}\lambda(\mathbf{x}', \mathbf{x}, c)\left[\mathbb{E}_{\epsilon,t}\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, \mathbf{x}, c) - \mathbf{x}'\|_2^2\right] \tag{6}$$

The whole process can be described as: we first sample a pair from dataset D, and then choose the corresponding specialist q_s to sample demonstrations $\mathbf{x}'$ for our editing model $\hat{\mathbf{x}}_\theta(\mathbf{x}_t, \mathbf{x}, c)$ to approximate with an importance weight of $\lambda(\mathbf{x}', \mathbf{x}, c)$. We formally provide the algorithm in 1. In our specialist-to-generalist framework, we need to have a series of specialist models $\{q_s(\cdot)\}_s$ and an importance function $\lambda(\cdot)$. We describe them separately in subsection 3.2 and subsection 3.3.

3.2 Constructing Specialist Models

We group the image editing task into 7 categories as summarized in Table 2. For each category, we train or build a task specialist $p_s(\mathbf{x}' \mid \mathbf{x}, c)$ to generate millions of examples. Table 2 provides detailed information on task groups and example editing instructions c. In this section, we briefly summarize each specialist, with details available in Appendix A.1.

Object Replacement. We trained an image-inpainting model to serve as the specialist $q_{\text{obj\_replace}}$ for object replacement. Given a image $\mathbf{x}$ and an object caption c_{obj} and a object mask M_{obj}. The $q_{\text{obj\_replace}}$ can fill the content indicated by the mask with an object in c_{obj}. We then generate an object replacement sample by masking out an existing object and fill the image with a new object.
Object Removal. We trained an image inpainting model to serve as the specialist $q_{\text{obj\_removal}}$ for object removal. We use a similar procedure as in the object replacement but use a predicted background content caption to inpaint the masked image.
Object Addition. We treat object addition as the inverse task of object removal. Specifically, for each pair of editing examples generated by the object removal specialist, we reverse the roles of the source and target images to create a new pair.
Attribute Modification. We adopt the Prompt-to-Prompt (P2P) (Hertz et al., 2023) pipeline to generate examples. To enable precise modification, we provide a mask M_{obj} for the object and force P2P to only make edits inside the mask.
Background Swap. We trained an image inpainting model to serve as the specialist $q_{\text{obj\_background\_swap}}$. We use a similar procedure as in the object replacement but use an inverse mask

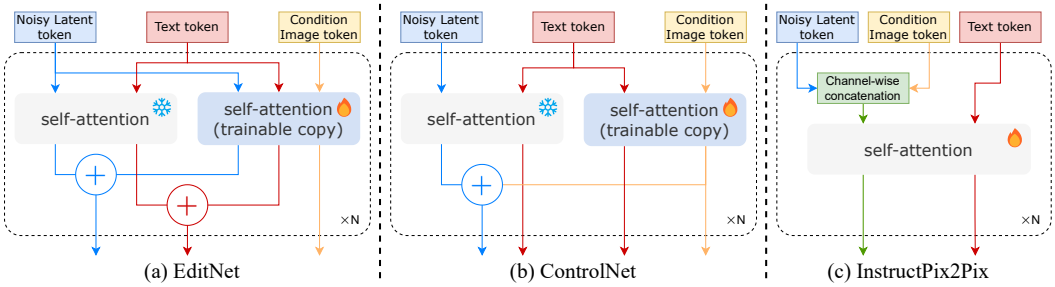

Figure 3: Architecture Comparison between **EditNet(ours)**, ControlNet and InstructPix2Pix for DiT models. Unlike ControlNet's parallel execution, EditNet allows adaptive adjustment of control signals by intermediate representations interaction between the control branch and the original branch. EditNet also updates the text representation, enabling better task understanding.

of the object to indicate the background and guide the inpainting.

Environment Modification. For environment modification, we use a P2P pipeline with an inverse mask of the object to indicate the background.

Style Transfer. We use CosXL-Edit (Boesel & Rombach, 2024) as the specialist model as its training data contains a large number of style transfering examples. We provide CosXL-Edit with $(\mathbf{x}, c)$, and let it generates the edited image $\mathbf{x}'$.

3.3 Importance Weighting

The importance weighting function λ takes as input a tuple of source image, edited image, and editing prompt. Its purpose is to assign higher weights to data points that are more likely to be sampled from the ground truth distribution, and lower weights to the unlikely ones. This is essentially a quality measure to up-weight high-quality samples. Unlike previous work, we do not use CLIP score because prior work (Jiang et al., 2024) has shown its low correlation with human judges. Instead, we propose to use large multimodal models (LMMs) to approximate the weighting function, as they demonstrate strong image understanding. Following VIEScore (Ku et al., 2024), we designed a prompting template for GPT-4o (Achiam et al., 2023) to evaluate the image editing pairs and output a score on a scale from 0 to 10. We then filter out data with a score greater than or equal to 9, so the LMM essentially serves as a binary weighting function:

$$\lambda(\mathbf{x}', \mathbf{x}, c) = \begin{cases} 1, & \text{if LMM}(\text{prompt}, \mathbf{x}', \mathbf{x}, c) \geq 9 \\ 0, & \text{otherwise} \end{cases}$$

Details of the prompt template are provided in the Appendix.

While the GPT-4o is an effective choice for this task, scoring large-scale datasets with millions of examples is extremely costly and time-consuming. Therefore, we employ knowledge distillation from GPT-4o to a smaller 8B model, InternVL2 (Chen et al., 2024b). For each task, we sample 50K data points and instruct GPT-4o to output both a score and a score rationale. We fine-tune internVL2 on these GPT-4o-generated examples. After fine-tuning, InternVL2 performs as an ideal scoring function due to its smaller size and efficiency. A comparison of the model's performance before and after fine-tuning is presented in the Appendix. Finally, we apply the fine-tuned InternVL2 model to filter data across a dataset with millions of samples, retaining only those with a score ≥ 9.

4 EditNet

We found that directly fine-tuning a pre-trained high-quality diffusion model like SD3 using image concatenation methods(Brooks et al., 2023) compromises the model's original representational capabilities (see Figure 6 and Section 5.2 for details comparison).

To enable a diffusion transformer to perform instruction-based image editing while preserving its original capabilities, we introduce **EditNet** to build Omni-Edit. EditNet can effectively transform common DIT models like SD3 into editing models.As illustrated in Figure 3, we replicate each layer of the original DIT block as a control branch. The control branch DIT blocks allow interaction between the original DIT tokens, conditional image tokens, and the editing prompts. The output of

the control branch tokens is then added to the original DIT tokens and editing prompts. Since the original DIT blocks are trained for generation tasks and are not aware of the editing instructions specifying which contents to modify and how to modify them, this design allows the control branch DIT to adjust the representations of DIT tokens and editing prompts according to the editing instruction, while still leveraging the strong generation ability of the DIT blocks. Compared to ControlNet (Zhang et al., 2023), our approach offers two key advantages that make it more suitable for image editing tasks: First, ControlNet does not update text representations, making it challenging to execute editing tasks based on instruction, particularly object removal, as it fails to understand removal intent(see Figure 5). Secondly, ControlNet's control branch operates in parallel without access to the original branch's intermediate representations. This fixed precomputation of control signals restricts the overall representation power of the network. We provide an ablation study on the OMNI-EDIT architecture design in Section5.2.

Figure 4: Qualitative comparison between baselines and OMNI-EDIT on a subset of the test set.

5 EXPERIMENTS

In this section, we first provide statistics of the OMNI-EDIT training set and test set. Then we introduce the human evaluation protocol and comparative baseline system. We present the main results in Section 5.1, highlighting the advantages of OMNI-EDIT in tacking multi-aspect ratio, multi-resolution, and multi-task image editing. In Section 5.2, we study the advantages of importance sampling for synthetic data. In Section 5.2, we perform an analysis to study the design of OMNI-EDIT.

OMNI-EDIT Training Dataset. We constructed the training dataset $\mathcal{D}$ by sampling high-resolution images with a minimum resolution of 1 megapixel from the LAION-5B (Schuhmann et al., 2022) and OpenImageV6 (Kuznetsova et al., 2020) databases. The images cover a range of aspect ratios including 1:1, 2:3, 3:2, 3:4, 4:3, 9:16, and 16:9. For the task of object swap, we employed a specialist model to generate 1.5 million entries. We then applied InternVL2 for importance weighting, retaining samples with scores of 9 or higher, resulting in a dataset of 150K entries for this task. Similarly, we generate around 1M samples for each task, then keep the top 10% as the final dataset. The final training dataset comprises 505K entries, with detailed information provided in Appendix 5.

OMNI-EDIT-Bench. To create a high-resolution, multi-aspect ratio, multi-task benchmark for instruction-based image editing, we manually collected 62 images from pexels (2024) and LAION-5B (Schuhmann et al., 2022). These images cover a variety of aspect ratios, including 1:1, 2:3, 3:2, 3:4, 4:3, 9:16, and 16:9. We ensured that the images feature a diverse range of scenes and object counts, from single to complex compositions. Additionally, we selected images with a relatively

Table 3: Main evaluation results on Omni-Edit-Bench. In each column, the highest score is bolded, and the second-highest is underlined.

Models	VIEScore (GPT4o)			VIEScore (Gemini)			Human Evaluation			
	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$Acc_{avg}\uparrow$
Inversion-based Methods										
DiffEdit	5.88	2.73	2.79	6.09	2.01	2.39	-	-	-	-
SDEdit	6.71	2.18	2.78	6.31	2.06	2.48	-	-	-	-
End-to-End Methods										
InstructPix2Pix	7.05	3.04	3.45	6.46	1.88	2.31	-	-	-	-
MagicBrush	6.11	3.53	3.60	6.36	2.27	2.61	-	-	-	-
UltraEdit(SD-3)	6.44	4.66	4.86	6.49	4.33	4.45	0.72	0.52	0.57	0.20
HQ-Edit	5.42	2.15	2.25	6.18	1.71	1.96	0.80	0.27	0.29	0.10
CosXL-Edit	<u>8.34</u>	<u>5.81</u>	<u>6.00</u>	<u>7.01</u>	<u>4.90</u>	<u>4.81</u>	<u>0.82</u>	<u>0.56</u>	<u>0.59</u>	<u>0.35</u>
HIVE	5.35	3.65	3.57	5.84	2.84	3.05	-	-	-	-
OMNI-EDIT	**8.38**	**6.66**	**6.98**	**7.06**	**5.82**	**5.78**	**0.83**	**0.71**	**0.69**	**0.55**
Δ - Best baseline	+0.04	+0.85	+0.98	+0.05	+0.92	+0.97	+0.01	+0.15	+0.10	+0.20

high aesthetic score to better align with the practical use cases of image editing. For each image, we tasked the model with performing 7 tasks as outlined in Table 2. This results in a total of 434 edits.

OMNI-EDIT implementation details. The OMNI-EDIT model is built upon Stable diffusion 3 (Esser et al., 2024) with EditNet architecture. The stable diffusion 3 has 24 DiT layers. Each layer has a corresponding EditNet layer. We train OMNI-EDIT on the 505K OMNI-EDIT training dataset for 2 epochs on a single node with 8 H100 GPUs.

Baseline models. We compare OMNI-EDIT with 8 other text-guided image editing baselines: MagicBrush (Zhang et al., 2024a), InstructPix2Pix (Brooks et al., 2023), UltraEdit(SD3) (Zhao et al., 2024), DiffEdit (Couairon et al., 2022), SDEdit (Meng et al., 2022), CosXL-Edit (Boesel & Rombach, 2024), HIVE (Zhang et al., 2024b) and HQ-Edit (Hui et al., 2024).

Evaluations Protocol We conduct both human evaluation and automatic evaluation. For the human evaluation, we follow the procedure from Ku et al. (2023) to rate in two criteria: Semantic Consistency (SC) and Perceptual Quality (PQ). Both scores are in $\{0, 0.5, 1\}$. For SC, the human subject is asked to rate the consistency between 1) the edited image and the editing instruction (whether the editing instruction is reflected on the edited image) and between 2) the source image and the edited image (whether the model makes the edit that is beyond the editing instruction). For PQ, the subject is asked to rate on the quality of edited image). We then calculate a overall score $O = \sqrt{SC \times PQ}$ that measures the overall quality of the edit. We also calculate the accuracy of the edit, which is defined by the percentage of $SC = 1$ among all examples. We recruit four human raters and require them to evaluate all the editing examples. For LMMs' evaluation, we follow the procedure from Ku et al. (2024) where models (in particular, we chose GPT4o and Gemini) are also asked to give SC and PQ scores but on a scale of 0-10. We then normalize the scale to 0-1.

5.1 MAIN RESULTS

We provide a qualitative comparison with baseline models in Figure 4. We show the top 4 baselines with OMNI-EDIT on a subset of the OMNI-EDIT-Bench. We provide more results in Fig 12 and 13. Our main results are detailed in Table 7, where we provide the VIEScore and conduct human evaluation on the Top2 baselines and OMNI-EDIT. In Figure 1, OMNI-EDIT demonstrates its capability to handle diverse editing tasks across various aspect ratios and resolutions. The results are notably sharp and clear, especially in the addition/swap task, where new content is seamlessly integrated. This underscores the effectiveness of the Edit-Net design in preserving the original image generation capabilities of the base text-image generative model. Similarly, in Figure 4, OMNI-EDIT uniquely adds a clean and distinct NASA logo onto a T-shirt. Table 7 corroborates this with OMNI-EDIT achieving the highest Perceptual Quality (PQ) score among the models evaluated.

We highlight the efficacy of our proposed specialist-to-generalist learning framework. Unlike baseline models that utilize a single method for generating synthetic data—often the prompt-to-prompt method—This method typically alters the entire image, obscuring task-specific data. In contrast, OMNI-EDIT leverages task-specific data curated by experts, resulting in a clearer task distribution and improved adherence to editing instructions. Both the VIEScore and human evaluations in Table 7 demonstrate that our method significantly outperforms the best baseline in following editing instructions accurately and minimizing over-editing. For instance, baseline models frequently misunderstand the task intent as illustrated in Figure 4, where the CosXL-Edit model fails to recognize the removal task and incorrectly interprets a bird addition as a swap between a panda and a bird.

Lastly, baseline models often produce blurry images on the OMNI-EDIT-Bench, as they are trained at resolutions limited to 512x512 or even 256x256, and they perform poorly on non-square aspect ratios. For example, with a 3:4 aspect ratio, the baselines struggle to perform editing. OMNI-EDIT, trained on data with multiple aspect ratios, maintains robust editing capabilities across the diverse aspect ratios encountered on the Omni-Bench, as evidenced in Figure 4.

5.2 ABLATION STUDY

In this section, We provide an ablation study w.r.t importance weighting and EditNet.

Ablation study on the importance sampling. We study a baseline that utilizes the same architecture as OMNI-EDIT, but instead of applying importance scoring and filtering, we sample 505K examples directly from the 5M pre-filtering dataset 5 and compare it with OMNI-EDIT. As shown in Table 4, we observe a significant decrease in VIEScores for both PQ and SC metrics.

Ablation Study on OMNI-EDIT Architecture Design. We conducted an analysis of OMNI-EDIT 's architectural design in comparison with two baseline models: OMNI-EDIT-controlnet and OMNI-EDIT-controlnet+textcontrol. OMNI-EDIT-controlnet is an adaptation of SD3-ControlNet trained on the OMNI-EDIT training dataset, while OMNI-EDIT-controlnet+textcontrol is a variant of SD3-ControlNet that, at each layer, not only integrates the image tokens from the ControlNet branch with the image tokens in the generative branch but also incorporates text tokens from the ControlNet branch alongside those in the generative text branch.

Table 4: Ablation on importance sampling and OMNI-EDIT architecture design.

Models	VIEScore (GPT4o)			VIEScore (Gemini)		
	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$
OMNI-EDIT	8.38	6.66	6.98	7.06	5.82	5.78
OMNI-EDIT- controlnet + textcontrol	6.45	4.70	4.89	6.50	4.35	4.48
OMNI-EDIT- controlnet	6.35	4.60	4.75	6.40	4.25	4.35
OMNI-EDIT w/o importance sampling	6.20	2.95	3.30	6.40	1.80	2.25

Our analysis, as shown in Figure 5, reveals that OMNI-EDIT-controlnet struggled to accurately capture task intent. This is primarily because the ControlNet branch does not update the text representation. For instance, in object removal tasks, prompts like "Remove ObjA" are common, yet the original DIT block remains unchanged, causing it to mistakenly generate an image of "ObjA." On the other hand, although OMNI-EDIT-controlnet+textcontrol successfully updates the text representation, it still encounters difficulties in content removal. The substantial VIEScores gap between OMNI-EDIT-controlnet+textcontrol and OMNI-EDIT in Table 4 underscores the importance of the intermediate representation interaction design in EditNet. We also compared OMNI-EDIT with the token concatenation method used in InstructPix2Pix. Token concatenation requires fine-tuning the entire network, which can distort the network's original representations. As illustrated in Figure 6, after fine-tuning on OMNI-EDIT training set, the representation of Batman is altered. In contrast, EditNet preserves the original representation of Batman while still learning the object swap task.

(a) Original Image (b) Edited by OmniEdit-controlnet (c) Edited by OmniEdit-controlnet+textcontrol (c) Edited by **Omni-Edit**

Figure 5: OMNI-EDIT-controlnet fails to grasp the task intent, while OMNI-EDIT-controlnet+textcontrol—a variant with a text-updating branch—recognizes the intent but struggles with content removal. In contrast, OMNI-EDIT accurately removes content.

6 RELATED WORK

Image Editing via Generation Editing real images according to specific user requirements has been a longstanding research challenge (Crowson et al., 2022; Liu et al., 2020; Zhang et al., 2023; Shi

(a) Original Image (b) Edited Image by SD3-Concat (c) Edited Image by Omni-Edit (d) Image Generated by SD3

Figure 6: (a) shows the source image. (d) presents images generated by SD3 in response to prompts for 'an upper body picture of Batman' and 'a shiny red vintage Chevrolet Bel Air car.' We use the prompts 'Replace the man with Batman' and 'Add a shiny red vintage Chevrolet Bel Air car to the right" to OMNI-EDIT and SD3-Concatenation, which was trained on OMNI-EDIT training data. From (b) and (c), one can observe that OMNI-EDIT preserves the generation capabilities of SD3, while SD3-Concatenation exhibits a notable degradation in generation capability.

et al., 2022; Ling et al., 2021). Since the introduction of large-scale diffusion models, such as Stable Diffusion (Rombach et al., 2022; Podell et al., 2024), significant progress has been made in tackling image editing tasks. SDEdit (Meng et al., 2022) introduced an approach that adds noise to the input image at an intermediate diffusion step, followed by denoising guided by the target text description to generate the edited image. Subsequent methods, such as Prompt-to-Prompt (Hertz et al., 2023) and Null-Text Inversion (Mokady et al., 2023), have focused on manipulating attention maps during intermediate diffusion steps for image editing. Other techniques like Blended Diffusion (Avrahami et al., 2022) and DiffEdit (Couairon et al., 2022) utilize masks to blend regions of the original image into the edited output. More recently, the field has seen a shift towards supervised methods, such as InstructP2P (Brooks et al., 2023), HIVE (Zhang et al., 2024b), and MagicBrush (Zhang et al., 2024a), which incorporate user-written instructions in an end-to-end framework. Our work follows this direction to develop end-to-end editing models without inversion.

Image Editing Datasets Due to the difficulty of collecting expert-annotated editing pairs, existing approaches rely heavily on synthetic data to train editing models. InstructP2P (Brooks et al., 2023) was the first to curate large-scale editing datasets using prompt-to-prompt filtering with CLIP scores. MagicBrush (Zhang et al., 2024a) subsequently improved data quality by incorporating a human-in-the-loop annotation pipeline based on DALLE-2. However, DALLE-2, primarily an inpainting-based method, struggles with global editing tasks such as style transfer and attribute modification. More recently, HQ-Edit (Hui et al., 2024) utilized DALLE-3 to curate editing pairs, although the source and target images lack pixel-to-pixel alignment, which is critical for preserving fine-grained details. Emu Edit (Sheynin et al., 2024) scaled up the training dataset to 10 million proprietary pairs, resulting in strong performance, but the lack of public access to their model checkpoints or API makes direct comparison difficult. UltraEdit (Zhao et al., 2024) proposed another inpainting-based approach, avoiding the use of DALLE-2 or DALLE-3 for data curation. However, like MagicBrush, it still faces limitations in handling complex global edits. Our work is the first to leverage multiple specialists to significantly expand the range of editing capabilities. Additionally, we are the first to use more reliable large multimodal models, for quality control in the editing process.

7 DISCUSSION

In this paper, we identify the imbalanced skills in the existing end-to-end image editing methods and propose a new framework to build more omnipotent image editing models. We surveyed the field and chose several approaches as our specialists to synthesize candidate pairs and adopt weighted loss to supervise the single generalist model. Our approach has shown significant quality boost across the broad editing skills. Throughout the experiments, we found that the output quality is highly influenced by the underlying base model. Due to the weakness of SD3, our approach is still not achieving its highest potential. In the future, we plan to use Flux or other more capable base models to see how much further we can reach with the current framework.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 18208–18218, 2022.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf, 2(3):8, 2023.

Frederic Boesel and Robin Rombach. Improving image editing models with generative data refinement. In The Second Tiny Papers Track at ICLR 2024, 2024.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18392–18402, 2023.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In Proceedings of the IEEE/CVF international conference on computer vision, pp. 9650–9660, 2021.

Junsong Chen, YU Jincheng, GE Chongjian, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In The Twelfth International Conference on Learning Representations, 2024a.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 24185–24198, 2024b.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. arXiv preprint arXiv:2210.11427, 2022.

Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. In European Conference on Computer Vision, pp. 88–105, 2022.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In Forty-first International Conference on Machine Learning, 2024.

Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. In The Twelfth International Conference on Learning Representations, 2024.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In The Eleventh International Conference on Learning Representations, 2023.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-prompt image editing with cross-attention control. In The Eleventh International Conference on Learning Representations, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020.

Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. Hq-edit: A high-quality dataset for instruction-based image editing. arXiv preprint arXiv:2404.09990, 2024.

Dongfu Jiang, Max Ku, Tianle Li, Yuansheng Ni, Shizhuo Sun, Rongqi Fan, and Wenhu Chen. Genai arena: An open evaluation platform for generative models. arXiv preprint arXiv:2406.04485, 2024.

Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. arXiv preprint arXiv:2403.06976, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. Advances in neural information processing systems, 35:26565–26577, 2022.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6007–6017, 2023.

Max Ku, Tianle Li, Kai Zhang, Yujie Lu, Xingyu Fu, Wenwen Zhuang, and Wenhu Chen. Imagenhub: Standardizing the evaluation of conditional image generation models. arXiv preprint arXiv:2310.01596, 2023.

Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. VIEScore: Towards explainable metrics for conditional image synthesis evaluation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 12268–12290, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.663. URL https://aclanthology.org/2024.acl-long.663.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. International journal of computer vision, 128(7):1956–1981, 2020.

Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. In Advances in Neural Information Processing Systems (NeurIPS), 2021.

Xihui Liu, Zhe Lin, Jianming Zhang, Handong Zhao, Quan Tran, Xiaogang Wang, and Hongsheng Li. Open-edit: Open-domain image manipulation with open-vocabulary instructions. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16, pp. 89–106. Springer, 2020.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In International Conference on Learning Representations, 2022.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6038–6047, 2023.

Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In ACM SIGGRAPH 2023 Conference Proceedings, pp. 1–11, 2023.

William S Peebles and Saining Xie. Scalable diffusion models with transformers. 2023 ieee. In CVF International Conference on Computer Vision (ICCV), volume 4172, 2022.

pexels. Pexels, 2024. URL www.pexels.com.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=di52zR8xgf.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In International conference on machine learning, pp. 8748–8763. PMLR, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10684–10695, 2022.

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. arXiv preprint arXiv:2403.12015, 2024.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems, 35:25278–25294, 2022.

Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8871–8879, 2024.

Yichun Shi, Xiao Yang, Yangyue Wan, and Xiaohui Shen. Semanticstylegan: Learning compositional generative priors for controllable image synthesis and editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11254–11264, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In International Conference on Learning Representations, 2021.

Linoy Tsaban and Apolinário Passos. Ledits: Real image editing with ddpm inversion and semantic guidance. arXiv preprint arXiv:2307.00522, 2023.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1921–1930, June 2023.

Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with natural language. arXiv preprint arXiv:2312.04965, 2023.

Fred Zhang. stable-diffusion-prompts-2.47m, 2024. URL https://huggingface.co/datasets/FredZhang7/stable-diffusion-prompts-2.47M?row=19.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. Advances in Neural Information Processing Systems, 36, 2024a.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3836–3847, 2023.

Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, et al. Hive: Harnessing human feedback for instructional visual editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9026–9036, 2024b.

Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. arXiv preprint arXiv:2407.05282, 2024.

A APPENDIX

Task	Pre-Filtering Number	After-Filtering Number
Object Swap	1,500,000	150,000
Object Removal	1,000,000	100,000
Object Addition	1,000,000	100,000
Background Swap	500,000	50,000
Environment Change	500,000	50,000
Style Transfer	250,000	25,000
Object Property Modification	300,000	30,000
Total	**5,050,000**	**505,000**

Table 5: Omni-Edit training dataset statistics reflecting the number of samples before and after importance scoring and filtering with o-score ≥ 9.

Algorithm 1 Specialist-to-Generalist Learning Framework

Require: Dataset $\mathcal{D} = \{(\mathbf{x_i}, c_i)\}_{i=1}^{N}$ of image-text instruction pairs
Require: $\mathcal{K}$ task specialist model q_k
Ensure: Generalist diffusion model parameterized by θ
1: **Initialize** a buffer $\mathcal{G} \leftarrow \emptyset$
2: **for** each pair of $\{(\mathbf{x_s}, c_s)\}$ in $\mathcal{D}$ **do**
3: $q_s = f(c_s)$, where $f : \mathcal{C} \rightarrow \mathcal{S}$ maps from the instruction space to the set of specialists.
4: $\mathbf{x'_s} \sim q_s(\mathbf{x'_s}|\mathbf{x_s}, c_s)$.
5: Compute importance weight $\lambda(\mathbf{x'_s}, \mathbf{x_s}, c_s)$
6: $\mathcal{G} \leftarrow \mathcal{G} \cup \{(\mathbf{x'_s}, \mathbf{x_s}, c_s), \lambda(\mathbf{x'_s}, \mathbf{x_s}, c_s)\}$
7: **end for**
8: Train generalist model θ on dataset $\mathcal{G}$ using Eq. 6

A.1 TRAINING DATA GENERATION DETAILS

A.1.1 OBJECT REPLACEMENT

We trained a dedicated image-inpainting model to serve as an expert for object replacement. However, **in the object replacement task, the new object often differs in shape from the original.** As illustrated in Figure 7, existing inpainting models heavily rely on accurate masks to generate the new object. In practice, producing exact masks for the new object is not feasible. Consequently, these models struggle to handle shape differences when filling the region.

To enhance the robustness of the inpainter, we introduce four mask augmentation strategies during training:

1. **Random Strokes**: Employ randomly generated brush-like patterns as masks to introduce diverse masking shapes.
2. **Connected Random Strokes**: Draw random brush-like patterns connected to the original object mask, simulating more complex and irregular mask shapes.
3. **Dilated Masking**: Expand the object mask using a randomly sized elliptical dilation kernel, capped at image width $\div$ 10, to simulate variability in mask size and shape.
4. **Bounding Box Masking**: Use the object's bounding box region as the mask.

These augmentations improve the model's adaptability to variations in object shapes, enabling it to handle diverse replacement scenarios effectively.

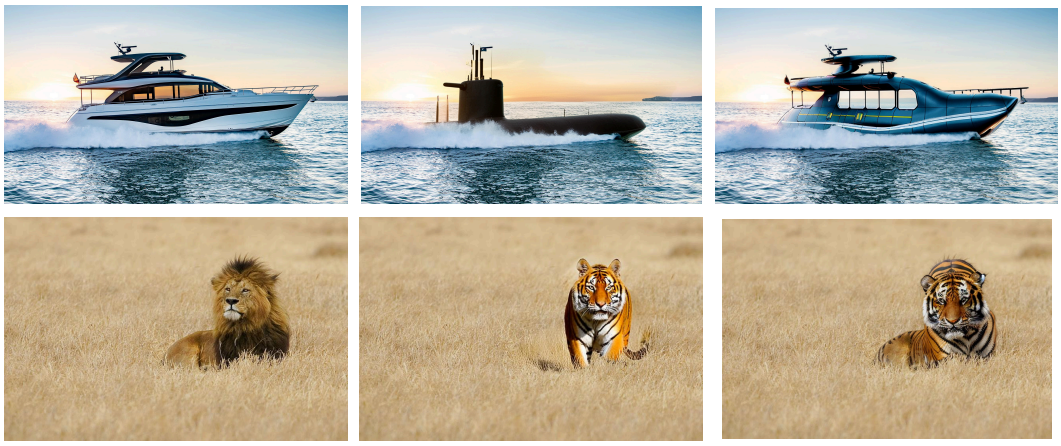

Figure 7: Comparison between the state-of-the-art (SOTA) inpainter and our proposed Object-Swap Expert using content prompts "**submarine**" and "**tiger**". The results demonstrate the effectiveness of our approach in handling object replacement with varying shapes.

Our inpainting model utilizes the BrushNet architecture (Ju et al., 2024), initialized with the Juggernaut-XL backbone for enhanced performance. The model is trained on the LAION-Aesthetic dataset (Schuhmann et al., 2022) for 800,000 steps using 8 A100 GPUs.

During training, given a source image $\mathbf{x}_{\text{src}}$ and an object caption C_{obj}, we employ GroundingDINO and SAM to generate an object mask M_{obj}. We then apply the four augmentation strategies mentioned above. The masked image is created by removing the object from the source image:

$$\mathbf{x}_{\text{masked}} = \mathbf{x}_{\text{src}} \odot (1 - M_{\text{obj}}) \tag{7}$$

Here, $\odot$ denotes element-wise multiplication, effectively masking out the object in $\mathbf{x}_{\text{src}}$. Both the mask M_{obj} and the object caption C_{obj} are provided as inputs to the expert model $q_{\text{obj\_replace}}$. The expert $q_{\text{obj\_replace}}$ is trained to reconstruct (inpaint) the original source image $\mathbf{x}_{\text{src}}$ from the masked image.

During inference, we sample 200K images from the LAION and OpenImages datasets, ensuring a diverse range of resolutions close to 1 megapixel. **To generate diverse and creative data for object replacement**, for each image, we utilize GPT-4o to propose five object replacement scenarios. Specifically, GPT-4o identifies five interesting source objects $C_{\text{src\_obj}}$ within the image and suggests corresponding target objects $C_{\text{trg\_obj}}$ for replacement.

For each proposed replacement, we perform the following steps:

1. Mask Generation: Use GroundingDINO and SAM to generate the object mask $M_{\text{src\_obj}}$ for the source object $C_{\text{src\_obj}}$.
2. Mask Dilation: Apply a dilation operation to $M_{\text{src\_obj}}$ to expand the mask boundaries.
3. Image Editing: Apply the expert model to generate the edited image $\mathbf{x}_{\text{edit}}$ by replacing the source object with the target object $C_{\text{trg\_obj}}$:

$$\mathbf{x}_{\text{edit}} = q_{\text{obj\_replace}}\left(\mathbf{x}_{\text{src}} \odot (1 - M_{\text{src\_obj}}),\ M_{\text{src\_obj}},\ C_{\text{trg\_obj}}\right) \tag{8}$$

In this equation:

- $\mathbf{x}_{\text{src}} \odot (1 - M_{\text{src\_obj}})$ represents the source image with the target object masked out.
- $M_{\text{src\_obj}}$ is the mask of the source object to be replaced.
- $C_{\text{trg\_obj}}$ is the caption of the target object for replacement.

Then a pair of instruction-based image editing examples will be: $\langle \mathbf{x}_{\text{src}}, \mathbf{x}_{\text{edit}}, T \rangle$. The instruction T initially just be "Replace $C_{\text{src\_obj}}$ with $C_{\text{trg\_obj}}$ ". We then employ large multimodal models (*LMMs*) to generate more detailed natural language instructions.

A.1.2 Object Removal

Existing object removal models present two main challenges: **Generative model-based object removers excel at generating natural fillings but often introduce new content**, even when provided with an empty prompt. In contrast, non-generative models like LaMa do not generate new content but struggle with removal quality.

Figure 8: Comparison between state-of-the-art (SOTA) object removers and our proposed Object-Removal Expert. Generative model-based object removers excel at generating natural fillings but often introduce new content, even when provided with an empty prompt. In contrast, non-generative models like LaMa do not generate new content but struggle with removal quality. Our proposed Object-Removal Expert strikes a balance between these approaches.

To address these limitations, we trained a dedicated image-inpainting model to serve as an expert for object removal. The inpainter is designed to fill arbitrary-shaped masks with plausible background content during inference. To achieve this, we introduce two new approaches:

1. Unique Training Process Tailored to Object Removal: During training, the inpainter is specifically trained to reconstruct random strokes. This encourages the model to generate realistic background textures rather than introducing new content during inference.

2. Guessing Background Content Using LMMs: Instead of feeding an empty prompt, we utilize LMMs to predict the likely background content after object removal. This further reduces the likelihood of generating new content.

Given a source image $\mathbf{x}_{\text{src}}$ and its corresponding caption C_{src}, we randomly apply strokes to create a mask M_{src}. The masked image is then generated as follows:

$$\mathbf{x}_{\text{masked}} = \mathbf{x}_{\text{src}} \odot (1 - M_{\text{src}})$$

Here, $\odot$ denotes element-wise multiplication, effectively masking out the object in $\mathbf{x}_{\text{src}}$. Both the mask M_{src} and the image caption C_{src} are provided as inputs to the expert model $q_{\text{obj\_removal}}$. The expert $q_{\text{obj\_removal}}$ is trained to reconstruct (inpaint) the original source image $\mathbf{x}_{\text{src}}$ from the masked image.

Our inpainting model adopts the BrushNet architecture (Ju et al., 2024), initialized with the Juggernaut-XL backbone for enhanced performance. The model is trained on the LAION-Aesthetic dataset (Schuhmann et al., 2022) for 600,000 steps using 8 A100 GPUs.

To generate training data for object removal, we sample 200K images from the LAION and Open-Images datasets, ensuring a diverse range of resolutions close to 1 megapixel. This dataset diversity enhances the robustness and generalizability of the model.

Although our trained inpainter does not tend to generate new content, we observed that directly using an empty prompt during inference leads to suboptimal results, sometimes producing uniform textures that do not naturally blend with the surrounding content. To address this, **we propose**

leveraging GPT-4 to predict the likely background content after object removal. This predicted background content is then used as a prompt for the inpainter. **This approach avoids generating new content and ensures the synthesis of natural and seamless background textures.**

For each image, we utilize GPT-4 to propose five objects to remove and predict the content of the space after removal. Specifically, GPT-4 identifies five interesting source objects $C_{\text{src\_obj}}$ within the image and predicts the new content after removing the object $C_{\text{trg\_background}}$.

For each proposed removal, we perform the following steps:

1. **Mask Generation**: Use GroundingDINO and SAM to generate the object mask $M_{\text{src\_obj}}$ for the source object $C_{\text{src\_obj}}$.
2. **Image Editing**: Apply the expert model to generate the edited image $\mathbf{x}_{\text{edit}}$ by infilling the masked region with the predicted background content $C_{\text{trg\_background}}$:

$$\mathbf{x}_{\text{edit}} = q_{\text{obj\_removal}}\left(\mathbf{x}_{\text{src}} \odot (1 - M_{\text{src\_obj}}),\ M_{\text{src\_obj}},\ C_{\text{trg\_background}}\right) \tag{9}$$

In this equation:

- $\mathbf{x}_{\text{src}} \odot (1 - M_{\text{src\_obj}})$ represents the source image with the target object masked out.
- $M_{\text{src\_obj}}$ is the mask of the source object to be removed.
- $C_{\text{trg\_background}}$ is the predicted content for the background after object removal.

Each instruction-based image editing example is represented as: $\langle \mathbf{x}_{\text{src}}, \mathbf{x}_{\text{edit}}, T \rangle$. Initially, the instruction T is set to "Remove $C_{\text{src\_obj}}$ from the image". We then employ large multimodal models (*LMMs*) to generate more detailed natural language instructions.

A.1.3 Object Addition

We conceptualize the object addition task as the inverse of the object removal process. Specifically, for each pair of editing examples generated by the object removal expert, we swap the roles of the source and target images to create a new pair tailored for object addition. This approach leverages the naturalness and artifact-free quality of the original source images, ensuring high-quality additions. Given a pair of editing examples $\langle \mathbf{x}_{\text{src\_removal}}, \mathbf{x}_{\text{edit\_removal}}, c_{\text{removal}} \rangle$ generated for object removal and $C_{\text{src\_obj\_removal}}$ represents the object to remove. We transform this pair into an object addition example by swapping $\mathbf{x}_{\text{src}}$ and $\mathbf{x}_{\text{edit}}$, and modifying the instruction accordingly. The resulting pair for object addition is $\langle \mathbf{x}_{\text{src}} = \mathbf{x}_{\text{edit\_removal}}, \mathbf{x}_{\text{edit}} = \mathbf{x}_{\text{src\_removal}}, c \rangle$, where c is the new instruction defined as "Add $C_{\text{src\_obj\_removal}}$ to the image."

A.1.4 Attribute Modification

We adapt the Prompt-to-Prompt (P2P) (Hertz et al., 2023; Sheynin et al., 2024) where pipeline where a text-guided image generation model is provided with a pair of captions $\langle C_{\text{src}}, C_{\text{edit}} \rangle$ and injects cross-attention maps from the input image generation to that during edited image generation. For example, a pair could be $\langle$"a blue backpack", "a purple backpack"$\rangle$ with the corresponding editing instruction "make the backpack purple".

However, as shown in the figure 9, the **P2P pipeline often alters the texture of the object and significantly changes the entire image, resulting in a very low success rate.** To address this issue, **we propose an approach that incorporates a mask during the P2P process**. This mask helps preserve the general shape and texture of the object while maintaining the background intact.

Specifically, to enable precise attribute modification of the desired object (e.g., the "backpack" in our example), we introduce an additional mask M_{obj} that isolates the object of interest. To create a pair of captions, we first obtain the source captions C_{src} from Zhang (2024). We then use GPT-4 to identify an object C_{obj} in the original caption C_{src}, propose an editing instruction to modify an attribute of C_{obj}, and output the edited caption C_{edit} with the object's updated attribute reflected.

We first let the image generation model PlaygroundV2.5 or Juggernaut-XL to generate a source image $\mathbf{x}_{\text{src}}$ using C_{src}. We then use GroundingDINO to extract mask M_{obj} that masks the object from the source image. We then apply P2P generation using PlaygroundV2.5 or Juggernaut-XL

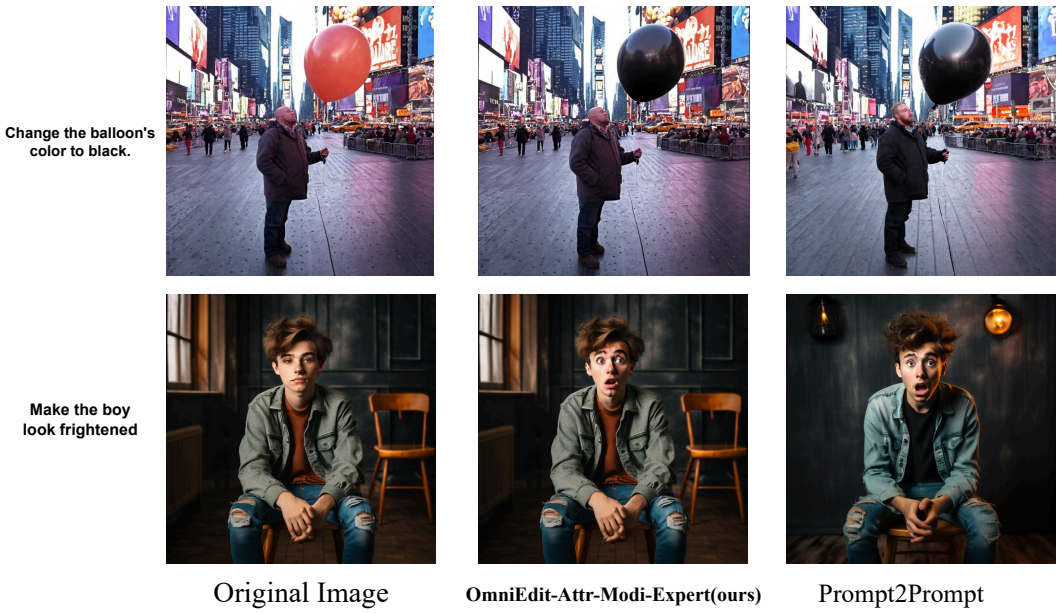

Figure 9: Comparison between the P2P pipeline and our proposed Expert model for attribute editing tasks. The P2P pipeline often alters the texture of the object and significantly changes the overall image, whereas our Expert model achieves precise editing while preserving the original image's integrity.

with caption pair $\langle C_{\text{src}}, C_{\text{edit}} \rangle$. During the generation, we use the mask to control precise image editing control. In particular, let $x_{\text{src},t}$ denote the noisy source image at step t and $x_{\text{edit},t}$ denote the noisy edited image at step t, we apply the mask and force the new noisy edited image at time t be $M_{\text{obj}} \odot x_{\text{edited},t} + (1 - M_{obj}) \odot x_{\text{input},t}$. In other words, we keep background the same and only edit the object selected.

A.1.5 Environment Modification

For environment modification, we use P2P pipeline to generate original and edited image. **To ensure structural consistency between two images, we apply a mask of the foreground to maintain details in the foreground while changing the background**. In particular, given a source image caption C_{src}, we use GPT4 to identify the foreground (e.g., an object or a human) and apply GroundingDINO to extract mask $M_{\text{foreground}}$. During the generation, let $\mathbf{x}_{\text{src},t}$ denote the noisy source image at step t and $\mathbf{x}_{\text{edit},t}$ denote the noisy edited image at t. We apply the mask so that the new noisy edited image at time t is $M_{\text{foreground}} \odot \mathbf{x}_{\text{src},t} + (1 - M_{\text{foreground}}) \odot \mathbf{x}_{\text{edit},t}$. We also set $\tau_{\text{env}} = 0.7$ so that this mask operation on noisy image is only applied at the first τ_{env} of all timesteps.

A.1.6 Background Swap

We trained an image inpainting model to serve as the specialist $q_{\text{obj\_background\_swap}}$. Similar to the object removal expert, the inpainter is specifically trained to reconstruct random strokes, encouraging the model to generate realistic background textures rather than new content during inference. The same inpainter model as in the object removal task is used as the expert for the background swap task.

To generate data for background swapping, we follow a procedure similar to that of object replacement but use an inverse mask of the object to indicate the background area to guide the inpainting. Specifically, we sample 200K images from the LAION and OpenImages datasets. For each image, we employ GPT-4 to propose five background replacement scenarios. GPT-4 identifies the main objects $C_{\text{src\_obj}}$ in the image and suggests five interesting backgrounds $C_{\text{trg\_back}}$ for replacement.

Human: You are a professional digital artist. You will have to evaluate the effectiveness of the AI-generated image(s) based on the given rules. You will have to give your output in this way (Keep your reasoning concise and short.):
{
"score" : [...],
"reasoning" : "..."
}
and don't output anything else.

Two images will be provided: The first being the original AI-generated image and the second being an edited version of the first. The objective is to evaluate how successfully the editing instruction has been executed in the second image. Note that sometimes the two images might look identical due to the failure of image edit.
From a scale 0 to 10:
A score from 0 to 10 will be given based on the success of the editing.
- 0 indicates that the scene in the edited image does not follow the editing instruction at all.
- 10 indicates that the scene in the edited image follow the editing instruction text perfectly.
- If the object in the instruction is not present in the original image at all, the score will be 0.

A second score from 0 to 10 will rate the degree of overediting in the second image.
- 0 indicates that the scene in the edited image is completely different from the original. - 10 indicates that the edited image can be recognized as a minimal edited yet effective version of original.
Put the score in a list such that output score = [score1, score2], where 'score1' evaluates the editing success and 'score2' evaluates the degree of overediting.

Editing instruction: <instruction>
<Image> Image_embed</Image>
<Image> Image_embed</Image>

Assistant:

Figure 10: Prompt for evaluating SC score.

A.1.7 Style Transfer

We use CosXL-Edit (Boesel & Rombach, 2024) as the expert style transfer model. We provide CosXL-Edit with $\langle \mathbf{x}_{\text{src}}, c \rangle$ and let it generates the edited image $\mathbf{x}_{\text{edited}}$.

A.1.8 Importance Sampling

We apply the importance sampling as described in Section 3.3. Example prompts that are provided to LMMs are shown in Figure 10 and 11. We compute the Overall score following (Ku et al., 2024) as the importance weight. After importance sampling, we obtain our training dataset described in Table 5.

A.2 Additional Evaluation Result

We present additional evaluation results. In Table 6, we compare Omni-Edit with specialist models of three tasks on Omni-Edit-Bench (other specialist models cannot take in input image). As is shown in the Table, Omni-Edit shows comparable performance as the specialist models on tasks that specialist models specialize.

Figure 12 shows additional comparisons between Omni-Edit other baseline models. We observe that Omni-Edit consistently outperforms other baselines.

Human: You are a professional digital artist. You will have to evaluate the effectiveness of the AI-generated image.
All the images and humans in the images are AI-generated. So you may not worry about privacy or confidentiality.
You must focus solely on the technical quality and artifacts in the image, and **do not consider whether the context is natural or not**.
Your evaluation should focus on:
- Distortions
- Unusual body parts or proportions
- Unnatural Object Shapes
Rate the image on a scale from 0 to 10, where:
- 0 indicates significant AI-artifacts.
- 10 indicates an artifact-free image.
You will have to give your output in this way (Keep your reasoning concise and short.):
{
"score": ...,
"reasoning": "..."
}
and don't output anything else.

<Image> Image_embed</Image>
<Image> Image_embed</Image>

Assistant:

Figure 11: Prompt for evaluating PQ score.

Table 6: Comparison between OMNI-EDIT and our specialist models.

	VIEScore (GPT4o)			VIEScore (Gemini)		
	PQ_{avg} ↑	SC_{avg} ↑	O_{avg} ↑	PQ_{avg} ↑	SC_{avg} ↑	O_{avg} ↑
Obj-Remove-Specialist	9.10	7.76	7.82	7.46	5.39	4.84
OMNI-EDIT	8.45	7.16	7.23	7.37	5.45	5.09
Obj-Replacement-Specialist	8.48	6.92	7.02	7.06	5.68	5.36
OMNI-EDIT	8.95	7.74	8.14	7.00	7.77	7.09
Style-Transfer-Specialist	8.08	7.47	7.37	7.97	6.61	6.76
OMNI-EDIT	7.98	5.77	6.16	8.24	5.24	6.08

A.3 OMNI-EDIT-BENCH EVALUATION BREAKDOWN

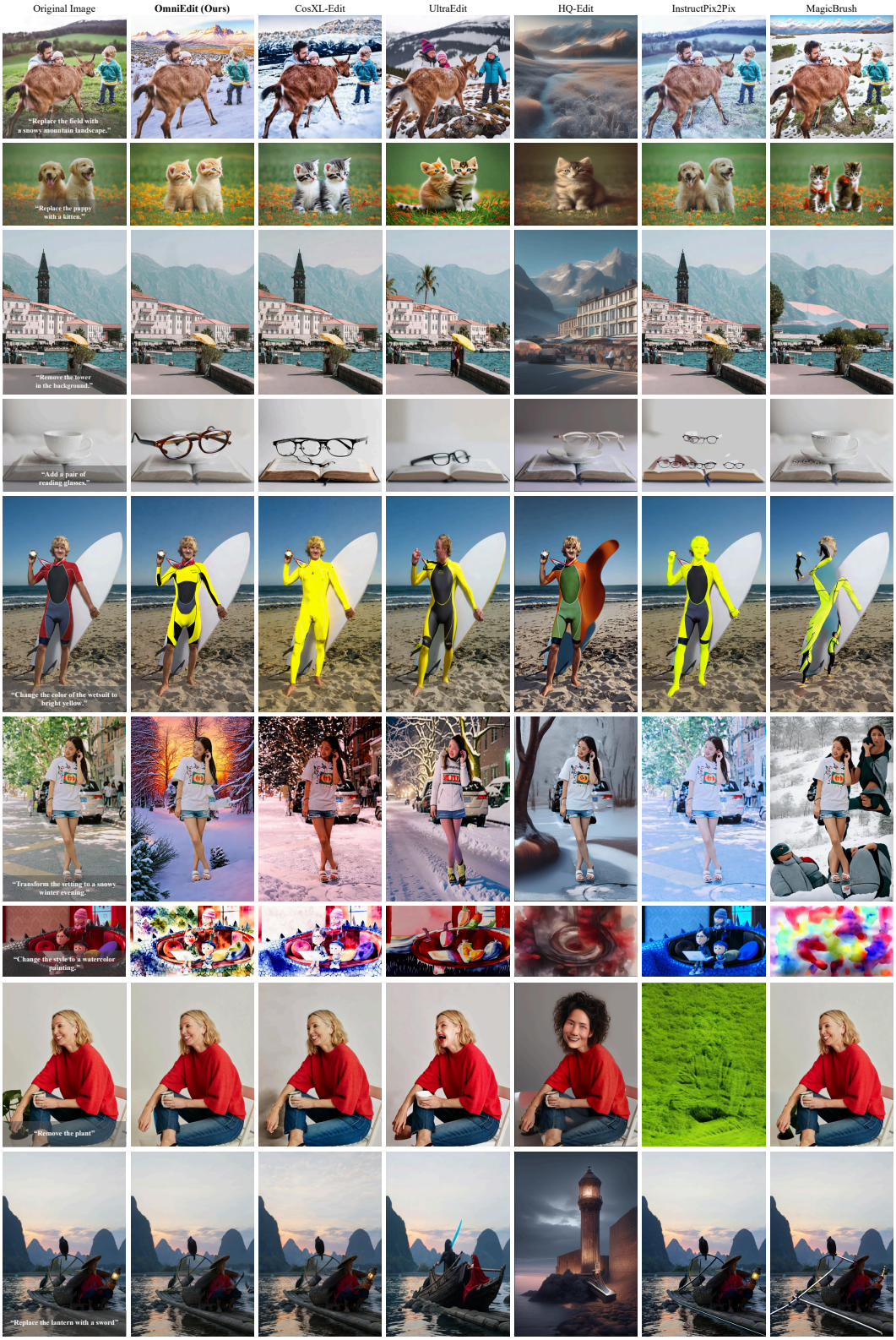

Figure 12: Additional qualitative comparisons between OMNI-EDIT and the baseline methods on OMNI-EDIT-BENCH

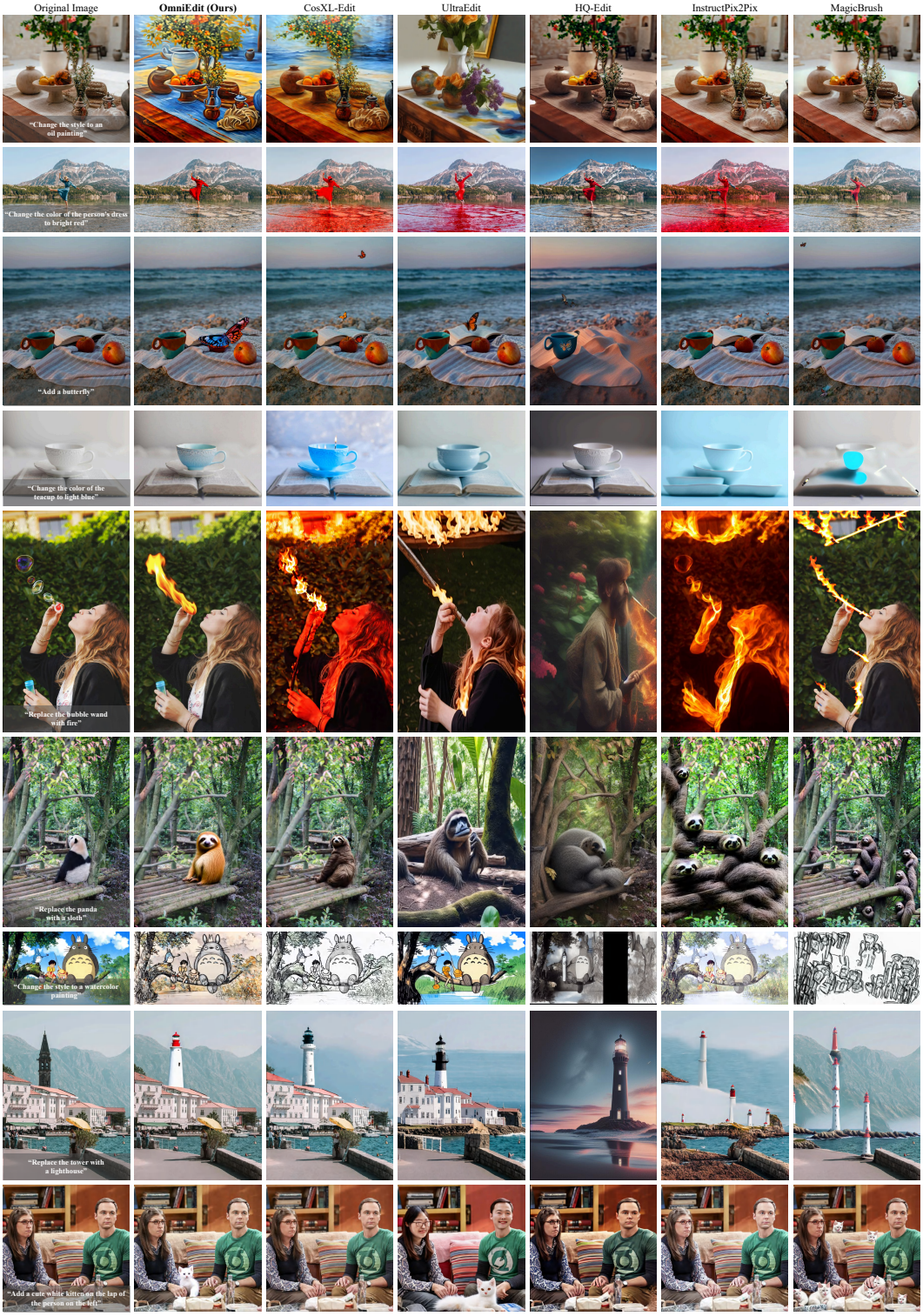

Figure 13: Additional qualitative comparisons between OMNI-EDIT and the baseline methods on OMNI-EDIT-BENCH

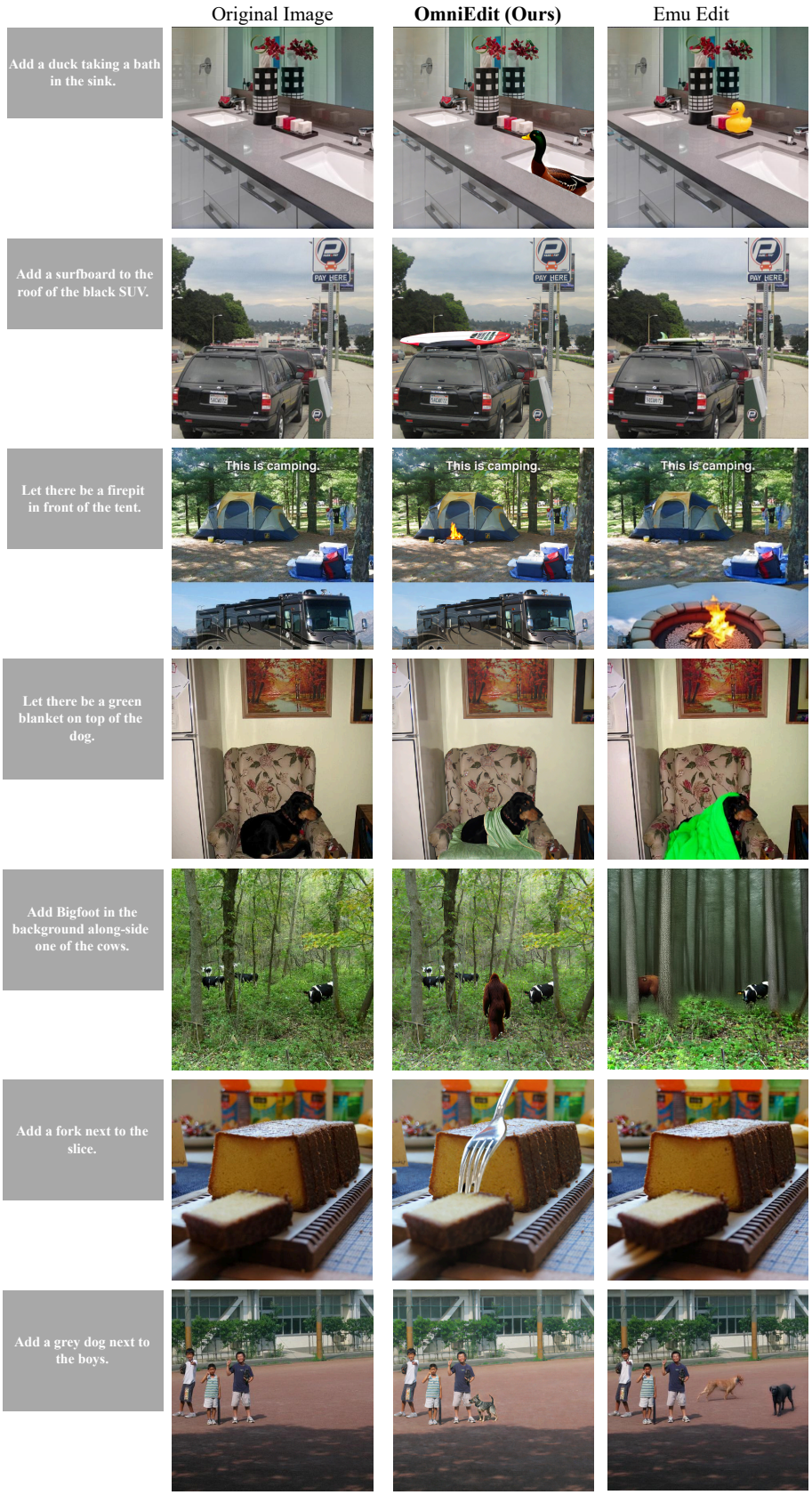

Figure 14: Qualitative comparisons between OMNI-EDIT and Emu Edit on Emu Edit test set.

Table 7: Main evaluation results on Omni-Edit-Bench. In each column, the highest score is bolded, and the second-highest is underlined.

Models	**VIEScore (GPT4o)**			**VIEScore (Gemini)**			**Human Evaluation**			
	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$Acc_{avg}\uparrow$
Inversion-based Methods										
DiffEdit	5.88	2.73	2.79	6.09	2.01	2.39	-	-	-	-
SDEdit	6.71	2.18	2.78	6.31	2.06	2.48	-	-	-	-
End-to-End Methods										
InstructPix2Pix	7.05	3.04	3.45	6.46	1.88	2.31	-	-	-	-
MagicBrush	6.11	3.53	3.60	6.36	2.27	2.61	-	-	-	-
UltraEdit(SD-3)	6.44	4.66	4.86	6.49	4.33	4.45	0.72	0.52	0.57	0.20
HQ-Edit	5.42	2.15	2.25	6.18	1.71	1.96	0.80	0.27	0.29	0.10
CosXL-Edit	<u>8.34</u>	<u>5.81</u>	<u>6.00</u>	<u>7.01</u>	<u>4.90</u>	<u>4.81</u>	<u>0.82</u>	<u>0.56</u>	<u>0.59</u>	<u>0.35</u>
HIVE	5.35	3.65	3.57	5.84	2.84	3.05	-	-	-	-
InstructDiffusion	5.18	4.89	4.11	6.57	4.68	4.56	-	-	-	-
OMNI-EDIT	**8.38**	**6.66**	**6.98**	**7.06**	**5.82**	**5.78**	**0.83**	**0.71**	**0.69**	**0.55**
Δ - Best baseline	+0.04	+0.85	+0.98	+0.05	+0.92	+0.97	+0.01	+0.15	+0.10	+0.20

Table 8: Evaluation results on Omni-Edit-Bench on object replacement task. In each column, the highest score is bolded, and the second-highest is underlined.

Models	**VIEScore (GPT4o)**			**VIEScore (Gemini)**			**Human Evaluation**			
	$PQ_{\text{avg}} \uparrow$	$SC_{\text{avg}} \uparrow$	$O_{\text{avg}} \uparrow$	$PQ_{\text{avg}} \uparrow$	$SC_{\text{avg}} \uparrow$	$O_{\text{avg}} \uparrow$	$PQ_{\text{avg}} \uparrow$	$SC_{\text{avg}} \uparrow$	$O_{\text{avg}} \uparrow$	$Acc_{\text{avg}} \uparrow$
Inversion-based Methods										
DiffEdit (SD-2.1)	5.65	3.87	3.77	5.82	3.05	3.40	-	-	-	-
SDEdit (SD-1.5)	6.66	3.03	3.56	6.10	2.45	2.91	-	-	-	-
End-to-End Methods										
InstructPix2Pix	6.15	2.21	2.34	6.24	1.24	1.62	-	-	-	-
MagicBrush	5.10	4.03	4.02	5.94	2.68	3.29	-	-	-	-
UltraEdit	6.35	4.48	4.81	6.21	4.35	4.60	0.71	0.50	0.56	0.14
HQ-Edit	5.34	2.16	2.45	6.32	1.81	2.27	0.93	0.29	0.32	0.14
CosXL-Edit	<u>8.29</u>	<u>5.79</u>	<u>5.86</u>	<u>6.89</u>	<u>5.34</u>	<u>5.09</u>	<u>0.93</u>	<u>0.79</u>	<u>0.85</u>	<u>0.57</u>
HIVE	4.92	3.98	3.82	5.45	3.15	3.09	-	-	-	-
InstructDiff	5.06	5.15	4.12	<u>6.89</u>	4.65	4.73	-	-	-	-
Omni-Edit	**8.95**	**7.74**	**8.13**	**7.00**	**6.77**	**6.09**	**0.93**	**0.86**	**0.87**	**0.71**
Δ - Best baseline	+0.66	+1.95	+2.27	+0.11	+1.43	+1.00	+0.00	+0.07	+0.02	+0.14

Table 9: Evaluation results on Omni-Edit-Bench on object removal task. In each column, the highest score is bolded, and the second-highest is underlined. CoXL-Edit shows better PQ as it often doesn't remove any object, resulting in the output image being identical to the input, as indicated by the low SC score.

Models	**VIEScore (GPT4o)**			**VIEScore (Gemini)**			**Human Evaluation**			
	$PQ_{\text{avg}} \uparrow$	$SC_{\text{avg}} \uparrow$	$O_{\text{avg}} \uparrow$	$PQ_{\text{avg}} \uparrow$	$SC_{\text{avg}} \uparrow$	$O_{\text{avg}} \uparrow$	$PQ_{\text{avg}} \uparrow$	$SC_{\text{avg}} \uparrow$	$O_{\text{avg}} \uparrow$	$Acc_{\text{avg}} \uparrow$
Inversion-based Methods										
DiffEdit (SD-2.1)	6.44	3.66	3.85	6.31	1.69	1.86	-	-	-	-
SDEdit (SD-1.5)	6.61	1.00	1.67	6.42	0.79	1.16	-	-	-	-
End-to-End Methods										
InstructPix2Pix	6.42	1.37	1.41	6.21	0.16	0.21	-	-	-	-
MagicBrush	8.08	4.35	4.36	6.84	2.16	2.19	-	-	-	-
UltraEdit	6.63	4.11	4.39	6.53	2.92	2.93	0.79	<u>0.50</u>	<u>0.59</u>	0.15
HQ-Edit	5.02	0.79	1.01	5.81	0.69	0.79	0.86	0.21	0.24	0.00
CosXL-Edit	**9.13**	2.39	2.58	**7.42**	0.16	0.13	**0.93**	0.29	0.29	<u>0.29</u>
HIVE	5.15	1.55	1.85	5.56	1.16	1.21	-	-	-	-
InstructDiff	5.69	<u>5.74</u>	<u>4.92</u>	7.14	<u>5.05</u>	<u>4.19</u>	-	-	-	-
Omni-Edit	<u>8.45</u>	**6.16**	**6.24**	<u>7.37</u>	**5.45**	**5.10**	<u>0.86</u>	**0.69**	**0.74**	**0.56**
Δ - Best baseline	$-(0.68)$	$+0.42$	$+1.32$	$-(0.05)$	$+0.40$	$+0.91$	$-(0.07)$	$+0.19$	$+0.15$	$+0.27$

Table 10: Results of object addition on Omni-Edit-Bench. In each column, the highest score is bolded, and the second-highest is underlined. CoXL-Edit shows better PQ as it often doesn't add any object, resulting in the output image being identical to the input, as indicated by the low SC score.

Models	**VIEScore (GPT4o)**			**VIEScore (Gemini)**			**Human Evaluation**			
	$PQ_{avg} \uparrow$	$SC_{avg} \uparrow$	$O_{avg} \uparrow$	$PQ_{avg} \uparrow$	$SC_{avg} \uparrow$	$O_{avg} \uparrow$	$PQ_{avg} \uparrow$	$SC_{avg} \uparrow$	$O_{avg} \uparrow$	$Acc_{avg} \uparrow$
Inversion-based Methods										
DiffEdit	5.53	2.87	2.93	5.84	2.03	2.41	-	-	-	-
SDEdit	6.50	1.21	1.91	5.85	1.34	1.60	-	-	-	-
End-to-End Methods										
InstructPix2Pix	7.19	2.65	2.88	6.84	1.50	1.87	-	-	-	-
MagicBrush	7.13	5.10	<u>5.54</u>	6.42	2.89	3.65	-	-	-	-
UltraEdit	6.85	4.77	5.17	6.37	<u>4.21</u>	4.29	**0.79**	<u>0.50</u>	<u>0.56</u>	<u>0.14</u>
HQ-Edit	5.08	1.77	1.82	6.27	1.63	1.83	0.64	0.29	0.32	0.00
CosXL-Edit	**7.77**	<u>5.11</u>	5.44	**6.90**	3.94	4.17	0.57	0.36	0.39	0.00
HIVE	4.84	3.10	3.30	5.44	1.97	2.26	-	-	-	-
InstructDiffusion	5.77	4.89	4.24	6.16	4.11	<u>4.31</u>	-	-	-	-
Omni-Edit	<u>7.35</u>	**6.03**	**6.40**	<u>6.66</u>	**5.79**	**5.45**	<u>0.71</u>	**0.64**	**0.67**	**0.43**
Δ - Best baseline	$-(0.42)$	$+0.92$	$+0.86$	$-(0.24)$	$+1.58$	$+1.14$	$-(0.08)$	$+0.14$	$+0.11$	$+0.29$

Table 11: Results for object property modification on Omni-Edit-Bench. In each column, the highest score is bolded, and the second-highest is underlined.

Models	**VIEScore (GPT4o)**			**VIEScore (Gemini)**			**Human Evaluation**			
	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$Acc_{avg}\uparrow$
Inversion-based Methods										
DiffEdit_SD_2_1	5.95	2.39	2.72	6.29	2.52	2.74	-	-	-	-
SDEdit_SD_1_5	6.66	1.26	1.77	6.00	1.18	1.38	-	-	-	-
End-to-End Methods										
InstructPix2Pix	7.73	2.69	3.76	6.48	2.81	3.31	-	-	-	-
MagicBrush	6.13	3.98	4.18	5.85	3.05	3.10	-	-	-	-
UltraEdit	6.58	3.94	4.56	6.47	4.27	4.40	0.64	0.36	0.42	0.00
HQ-Edit	5.18	2.13	2.27	5.84	1.79	1.82	**0.79**	0.21	0.24	0.10
CosXL-Edit	7.90	5.32	5.73	**6.66**	**6.00**	**5.69**	0.63	**0.43**	0.42	0.00
HIVE	5.53	3.18	3.40	5.66	2.71	2.99	-	-	-	-
InstructDiff	6.18	4.82	5.02	6.18	5.21	4.69	-	-	-	-
Omni-Edit	**8.56**	**5.74**	**6.23**	6.63	5.08	4.93	0.64	**0.43**	**0.43**	**0.14**
Δ - Best baseline	+0.66	+0.42	+0.50	−(0.03)	−(0.92)	−(0.76)	−0.15	+0.00	+0.01	+0.04

Table 12: Evaluation results for background replacement on Omni-Edit-Bench. In each column, the highest score is bolded, and the second-highest is underlined.

Models	**VIEScore (GPT4o)**			**VIEScore (Gemini)**			**Human Evaluation**			
	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$Acc_{avg}\uparrow$
Inversion-based Methods										
DiffEdit	6.35	1.19	1.53	6.16	0.82	1.23	-	-	-	-
SDEdit	7.05	2.34	3.05	6.19	2.42	2.80	-	-	-	-
End-to-End Methods										
InstructPix2Pix	5.92	2.52	2.73	6.24	1.74	2.21	-	-	-	-
MagicBrush	5.35	3.69	3.38	6.27	2.74	2.99	-	-	-	-
UltraEdit	6.19	4.76	4.60	6.52	4.81	4.72	0.64	0.57	0.60	0.29
HQ-Edit	6.10	2.16	2.34	6.34	1.60	1.76	**0.79**	0.36	0.42	0.14
CosXL-Edit	8.32	7.11	7.21	6.81	6.03	5.82	**0.79**	0.57	0.63	0.29
HIVE	5.34	3.06	2.95	5.84	2.16	2.60	-	-	-	-
InstructDiffusion	4.27	4.32	3.61	6.40	5.03	4.86	-	-	-	-
Omni-Edit	**8.53**	**7.45**	**7.76**	**7.05**	**6.03**	**6.18**	0.67	**0.86**	**0.69**	**0.71**
Δ - Best baseline	+0.21	+0.34	+0.55	+0.24	0.00	+0.36	-(0.12)	+0.29	+0.06	+0.42

Table 13: Evaluation results on global environment modification subcategory. In each column, the highest score is bolded, and the second-highest is underlined.

Models	**VIEScore (GPT4o)**			**VIEScore (Gemini)**			**Human Evaluation**			
	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$Acc_{avg}\uparrow$
Inversion-based Methods										
DiffEdit_SD_2_1	6.26	1.68	2.00	6.13	1.58	2.10	-	-	-	-
SDEdit_SD_1_5	7.66	3.47	4.24	6.73	3.89	4.39	-	-	-	-
End-to-End Methods										
InstructPix2Pix	8.02	3.65	4.49	**6.73**	1.90	2.75	-	-	-	-
MagicBrush	5.05	2.53	2.41	5.74	1.79	2.00	-	-	-	-
UltraEdit	6.82	5.73	5.73	6.19	5.81	5.60	0.57	0.57	0.56	0.29
HQ-Edit	5.97	2.58	2.66	6.19	2.15	2.47	0.79	0.43	0.42	0.29
CosXL-Edit	8.65	7.21	7.56	6.68	6.32	6.04	**0.93**	**0.86**	**0.89**	**0.71**
HIVE	6.24	4.89	4.67	6.29	4.26	4.49	-	-	-	-
InstructDiff	5.05	3.19	2.79	6.04	3.97	4.19	-	-	-	-
Omni-Edit	**8.85**	**7.73**	**7.95**	6.70	**6.40**	**6.09**	0.86	0.81	0.86	0.67
Δ - Best baseline	+0.20	+ (0.52)	+ (0.39)	- (0.03)	+ 0.08	+ 0.05	- (0.07)	- (0.05)	- (0.03)	- (0.04)

Table 14: Results on global style transfer in Omni-Edit-Bench. In each column, the highest score is bolded, and the second-highest is underlined. CoXL-Edit is the expert model to generate training data for OmniEdit in this task, and OmniEdit's performance is comparable to that of CoXL-Edit.

Models	**VIEScore (GPT4o)**			**VIEScore (Gemini)**			**Human Evaluation**			
	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$PQ_{avg}\uparrow$	$SC_{avg}\uparrow$	$O_{avg}\uparrow$	$Acc_{avg}\uparrow$
Inversion-based Methods										
DiffEdit_SD_2_1	5.00	3.45	2.73	6.44	2.40	3.08	-	-	-	-
SDEdit_SD_1_5	5.81	2.97	3.23	6.89	2.39	3.13	-	-	-	-
End-to-End Methods										
InstructPix2Pix	7.92	<u>6.19</u>	6.52	6.63	3.79	4.21	-	-	-	-
MagicBrush	5.95	1.03	1.33	7.58	0.61	1.09	-	-	-	-
UltraEdit	5.65	4.85	4.78	7.13	3.97	4.65	<u>0.79</u>	**0.64**	<u>0.64</u>	<u>0.43</u>
HQ-Edit	5.26	3.47	3.23	6.45	2.34	2.76	<u>0.79</u>	0.07	0.10	0.00
CosXL-Edit	<u>8.10</u>	**7.73**	**7.60**	<u>7.74</u>	**6.50**	**6.70**	**0.93**	**0.64**	**0.67**	**0.57**
HIVE	5.44	5.79	5.00	6.65	4.48	4.74	-	-	-	-
InstructDiff	4.23	5.11	3.98	7.18	4.77	4.77	-	-	-	-
Omni-Edit	**8.18**	<u>6.19</u>	<u>6.53</u>	**8.24**	<u>5.24</u>	<u>6.08</u>	**0.93**	**0.64**	<u>0.64</u>	**0.57**
Δ - Best baseline	$+(0.08)$	$-(1.96)$	$-(1.44)$	+0.50	$-(1.26)$	$-(0.62)$	0.00	0.00	$-(0.03)$	0.00