EditCLIP: Representation Learning for Image Editing

Qian Wang Aleksandar Cvejic Abdelrahman Eldesokey Peter Wonka
KAUST, Saudi Arabia

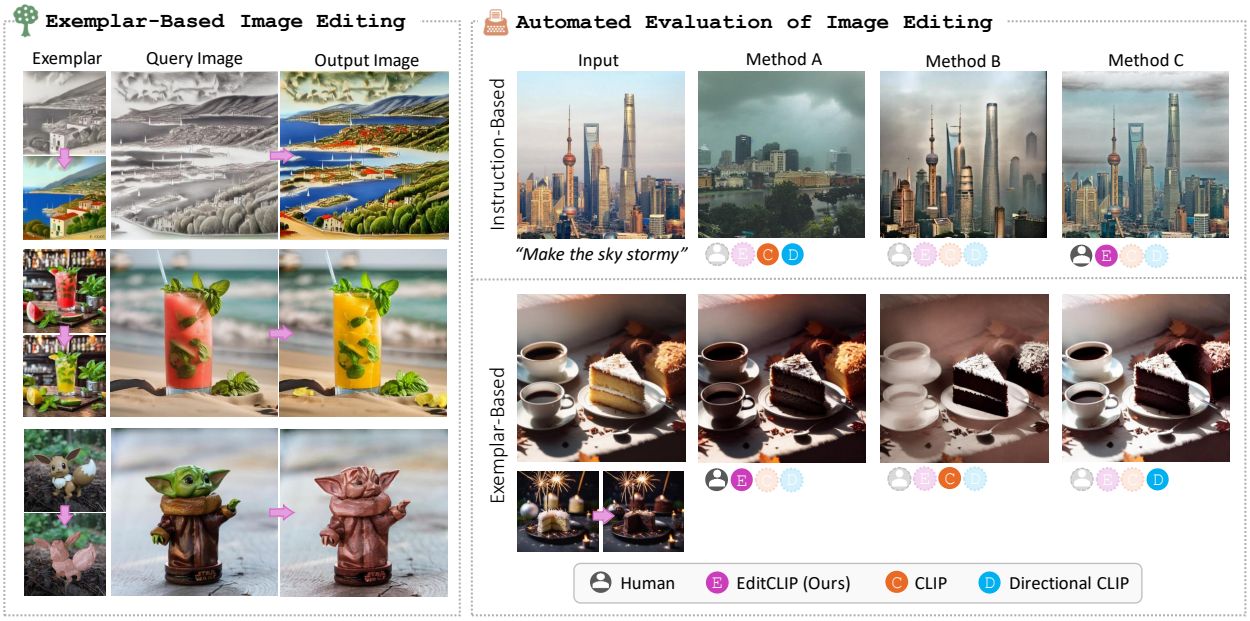

Figure 1. EditCLIP provides a unified representation of image edits by encoding the transformation between an image and its edited counterpart within the CLIP space. We demonstrate the effectiveness of EditCLIP embeddings in exemplar-based image editing and automated evaluation of image editing pipelines, where it achieves better alignment with human assessment.

Abstract

We introduce EditCLIP, a novel representation-learning approach for image editing. Our method learns a unified representation of edits by jointly encoding an input image and its edited counterpart, effectively capturing their transformation. To evaluate its effectiveness, we employ EditCLIP to solve two tasks: exemplar-based image editing and automated edit evaluation. In exemplar-based image editing, we replace text-based instructions in InstructPix2Pix [4] with EditCLIP embeddings computed from a reference exemplar image pair. Experiments demonstrate that our approach outperforms state-of-the-art methods while being more efficient and versatile. For automated evaluation, EditCLIP assesses image edits by measuring the similarity between the EditCLIP embedding of a given image pair and either a textual editing instruction or the EditCLIP embedding of another reference image pair. Experiments show that EditCLIP aligns more closely with human judgments than existing CLIP-based metrics, providing a reliable measure of edit quality and structural preservation. The code and model weights are available at https://github.com/QianWangX/EditCLIP.

1. Introduction

Image editing is a fundamental task in creative domains such as design and digital art, enabling creators to iteratively refine their creations to align with their artistic vision. Recent advancements in diffusion models [12, 28, 34–36] have revolutionized image editing [3–5, 17, 20, 23, 27, 37, 45, 46], leveraging their deep semantic understanding of images and artistic styles to apply highly realistic edits. Traditionally, diffusion-based editing approaches rely on textual instructions to specify the desired edits. Then, the internal dynamics of a diffusion model are manipulated to localize regions of interest and apply the edit. While effective,

instruction-based editing is limited by the diffusion model's understanding of language and the inherent limitations of natural language in describing complex edits, *e.g.* artistic styles with no established name and compound edits.

Several research directions have emerged to tackle these challenges, either by enhancing the semantic understanding of diffusion models to enable more complex and fine-grained edits [2, 4, 10, 14, 16, 22, 23, 26, 46] or by incorporating visual prompts [29, 32, 38, 48, 53] as a conditioning signal for diffusion models to perform exemplar-based editing. However, these research efforts face two major bottlenecks. First, they still rely on text to specify the edits. Even when visual exemplars are provided, they are ultimately mapped to a textual space either through Vision-Language Models (VLMs) [29, 38] or by optimizing special textual tokens based on the exemplars [32, 48].

Second, the evaluation of these approaches heavily relies on CLIP-based metrics [25, 33], which either measure the alignment between the edited image and the textual descriptions or compute a directional embedding vector between the original and edited images. However, these metrics primarily assess whether the edit is applied, disregarding whether the structure of the edited image deviates significantly from the original. Due to this limitation, researchers often rely on human evaluations through user studies to assess edit quality, which incurs higher costs and longer evaluation times.

We propose EditCLIP, a novel representation-learning approach for image editing that addresses these challenges altogether by learning an implicit representation of edits beyond linguistic constraints. Inspired by CLIP's ability to capture semantic relationships between images and texts, our method models the semantic relationships between image edits and their corresponding editing instructions within the CLIP space. Specifically, our model learns a unified representation of edits by encoding how reference images are transformed into their edited counterparts in relation to the provided instruction. We demonstrate the effectiveness of our model on two tasks: *exemplar-based image editing* and *automated evaluation of image editing* tasks.

In exemplar-based editing, given a single example of an image and its edited counterpart, our EditCLIP embedding is computed and used to guide the diffusion process to replicate the edit on a new output image without requiring textual editing instruction. This capability enables complex and precise edits, where describing the edit in natural language is challenging. For instance, an artist who applies multiple edits to an image but struggles to describe them in words can use EditCLIP to capture and transfer the edits seamlessly. Experiments show that our approach outperforms existing exemplar-based image editing methods across different types of edits and even outperforms the recent state-of-the-art approach InstaManip [29] despite having only 5.9% the number of parameters.

For automated evaluation of image editing, we measure the edit-instruction alignment by computing the similarity between the EditCLIP embedding of a given image pair and either the embedding of the textual editing instruction or the EditCLIP embedding of another reference image pair. Unlike CLIP-based metrics that independently embed images and compute differences between their global visual embeddings, EditCLIP embeddings directly capture how the image is transformed, taking into consideration how the edit is applied and if the unedited regions are preserved. Experiments show that EditCLIP aligns more closely with human judgments than existing CLIP-based metrics, providing a scalable and automated alternative for evaluating image editing methods. By streamlining evaluation, our approach can help accelerate the research of image editing.

Our contributions can be summarized as follows:

- We propose EditCLIP, a representation-learning approach that produces a unified representation for various types of image edits.
- We show that the learned representations can be used for exemplar-based image editing, replacing text-based instructions in diffusion models.
- We further show that EditCLIP provides a reliable edit representation, enabling the assessment of both edit quality and faithfulness to the reference image.

2. Related Work

2.1. Diffusion-Based Image Editing

The emergence of image diffusion models has driven the development of powerful image editing approaches, leveraging their deep understanding of image semantics. One category of these approaches is training-free, which either manipulates the internal representations of the diffusion U-Net [1, 5, 9, 20, 30, 39, 47], or manipulates the diffusion trajectory [3, 19, 23, 44] to achieve the desired edits. Another category fine-tunes a pre-trained diffusion model on image editing datasets, enabling it to apply edits [4, 18, 24, 50]. Alternatively, test-time optimization was employed in [8, 11, 31, 40, 41] to perform customized edits given a single image. In all these approaches, the textual embedding of an editing instruction serves as a condition to steer the diffusion model toward the intended edit.

2.2. Exemplar-Based Image Editing

A major limitation of instruction-based editing approaches is their reliance on language to describe the edit, which can be challenging for complex edits, especially when multiple edits are combined. Exemplar-based image editing addresses this issue by performing edits based on a user-provided reference image pair. Prior work [43] aimed at solving in-context learning tasks but could also handle

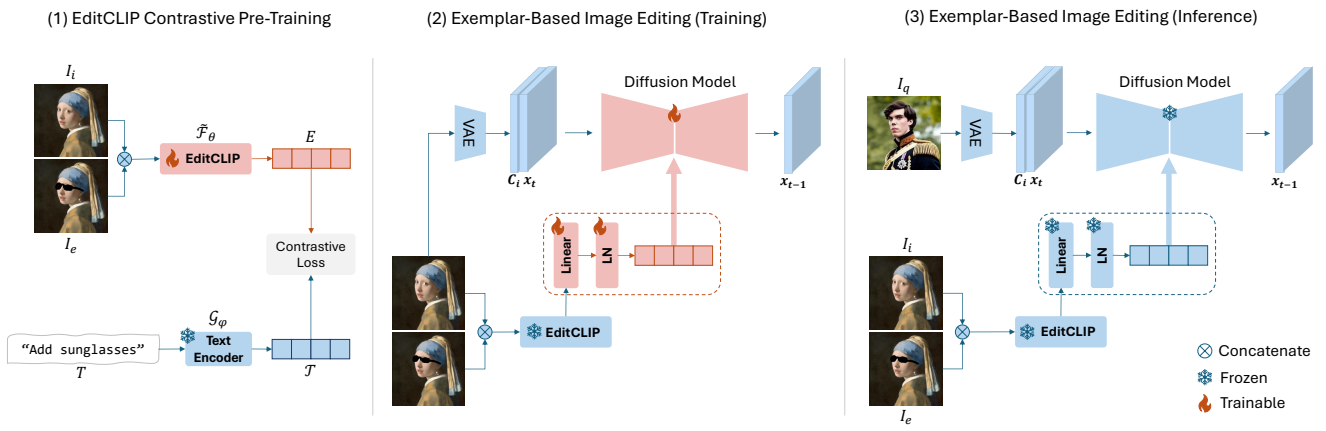

Figure 2. An overview of our proposed approach. EditCLIP is pre-trained similarly to CLIP, but the visual encoder processes a concatenated exemplar image pair. After pre-training, EditCLIP can replace the text encoder in InstructPix2Pix [4] to enable exemplar-based editing.

exemplar-based image editing, by projecting reference image embeddings into a ControlNet [51]. Approaches such as [32, 48] encoded the edit by optimizing special textual tokens derived from the reference image pair, which can then be used to apply the edit to new query images. Nonetheless, these methods are mostly limited to stylistic edits, and they may fail when there is more disparity between the exemplar and the query image, restricting their applicability to more diverse editing tasks.

Other works [29, 38] attempted to leverage Vision-Language Models (VLMs) to describe the edit between image pairs. Similarly, [53] optimized instruction pairs to represent the transformation between the exemplar pair. However, these approaches remain constrained by linguistic descriptions and introduce significant computational overhead due to the complexity of VLMs or the need for costly optimizations. In contrast, our proposed EditCLIP produces an implicit representation of edits, making it unconstrained by linguistic descriptions. This allows it to better capture complex edits that are difficult to express in natural language. Moreover, EditCLIP serves as a plug-and-play substitute for the CLIP text encoder in diffusion models, making it seamlessly integrated into popular editing pipelines such as InstructPix2Pix [4].

2.3. Evaluating Image Editing Approaches

A key aspect when developing image editing approaches is the evaluation protocol. A common practice is to use CLIP score [33] between the image embedding of the edited image and the text embedding of the editing instruction. However, this approach does not account for how much the edited image deviates from the original. To address this, previous works [25, 32] have proposed a directional CLIP score (CLIP directional similarity), which compares the directional embedding between the source and edited image with either the editing instruction or a directional embedding from a reference editing pair [32]. Nonetheless, these metrics only focus on how the edit is globally applied and do not take into account if the structure of the source image is preserved. In contrast, our proposed EditCLIP explicitly encodes the difference between the source and edited image, *i.e.*, the edit itself, effectively capturing both the edit and the deviation from the source image for a more accurate and detailed evaluation.

3. Method

We aim to design a representation learning approach for image editing, where edits can be implicitly encoded within an embedding space. Below, we introduce EditCLIP, a model designed to learn a general representation of edits. We first describe our approach and then analyze how our model captures edit semantics. Then, we explain how EditCLIP can be used for exemplar-based image editing as an alternative to text-based editing instructions. Finally, we demonstrate the versatility of EditCLIP embeddings by employing them for automated evaluation of image editing pipelines.

3.1. Representation Learning for Image Editing

In image editing, given an input image $I_i \in \mathbb{R}^{H\times W\times C}$, the objective is to produce an edited image $I_e \in \mathbb{R}^{H\times W\times C}$ based on a textual instruction T. This transformation can be formulated as:

$$I_e = \mathcal{U}(I_i; T) \ , \tag{1}$$

where $\mathcal{U}$ represents the editing pipeline that modifies I_i according to T. A key challenge lies in determining the level of detail required in the textual instruction T to achieve the intended edit. Ideally, T should specify how every element of I_i is transformed into I_e, but this is sometimes infeasible. Instead, an effective approach should aim to capture

the transformation from I_i to I_e in a more structured and learnable manner.

This problem shares similarities with representation learning of images and text in CLIP [33], where the goal was to learn a shared representation of images and text. CLIP has been shown to effectively capture semantic relationships between images and text from relatively coarse textual descriptions using contrastive learning on large-scale image-text pairs. Following this strategy, we aim to learn the semantics of edits within the CLIP space, leveraging its ability to encode meaningful transformations from textual guidance.

3.2. EditCLIP Pre-Training

A standard CLIP model consists of a visual encoder $\mathcal{F}_\theta$ and a text encoder $\mathcal{G}_\phi$, parameterized by learnable parameters θ and ϕ, respectively. Our objective is for the visual encoder to capture how the input image I_i is semantically and visually transformed into the edited image I_e. To achieve this, we modify the visual encoder $\mathcal{F}_\theta$ to accept a composite input image, where the input and edited images are concatenated along the channel dimension. We denote this new encoder as $\tilde{\mathcal{F}}$ and it produces an edit embedding E as:

$$E = \tilde{\mathcal{F}}_\theta(\texttt{concat}(I_i, I_e)) \in \mathbb{R}^{d_e \times 768}, \tag{2}$$

where d_e is the number of tokens for E. For the text encoder, we encode the editing instruction T into textual embedding $\mathcal{T}$, which describes the transformation from I_i to I_e, as:

$$\mathcal{T} = \mathcal{G}_\phi(T) \in \mathbb{R}^{d_t \times 768} \ , \tag{3}$$

where d_t is the number of tokens for $\mathcal{T}$. Following the contrastive learning paradigm of CLIP, we align the learned *editing space* with the textual space, where we train only the visual encoder while keeping the pre-trained textual encoder frozen. The training data is sampled from existing instruction-based image editing benchmarks, which provide triplets consisting of an input image I_i, its edited counterpart I_e, and the corresponding editing instruction T.

To analyze what the EditCLIP model learns, we follow the common practice of visualizing the attention of the $[CLS]$ token from the last attention head in the final transformer layer of the visual encoder [6]. As shown in Figure 3, EditCLIP focuses on the regions corresponding to the applied edits, such as shifting attention to the woman's torso and the edited cat on the right.

3.3. EditCLIP for Exemplar-Based Image Editing

To demonstrate the effectiveness of our proposed EditCLIP embeddings, we employ them as a substitute for textual editing instructions in Instruct-Pix2Pix (IP2P) [4]. IP2P is a diffusion-based image editing approach that conditions on a textual editing prompt and an input image to generate an edited output that fulfills the specified edit. To train IP2P with our embeddings, we feed the input image I_i and its edited counterpart I_e into EditCLIP to obtain the edit embedding E, as per Equation (2). The same input image is also encoded into the latent space of the diffusion model using the VAE encoder $\mathcal{E}$ that is concatenated with the input noise x_t.

To align E with the text embedding space originally used to train the diffusion model, we process it through a trainable linear layer followed by Layer Normalization. Note that we use the last hidden state of the EditCLIP visual encoder instead of the projected embedding, but we use E for simplicity. Finally, the diffusion model is fine-tuned using the standard diffusion noise-prediction loss to learn to denoise the latent of the edited image:

$$\mathcal{L}_{\text{noise}} = \mathbb{E}_{\mathcal{E}(I_e),\, \mathcal{E}(I_i),\, E,\, \epsilon \sim \mathcal{N}(0,1),\, t} \left[\|\epsilon - \epsilon_\theta(x_t, t, \mathcal{E}(I_i), E)\|_2^2 \right]. \tag{4}$$

where ϵ is the groundtruth noise added to the noisy latent x_t. The training pipeline is illustrated in Figure 2.

To further preserve the layout from the input image, we adopt an LPIPS loss [52] between the input image and the reconstructed image I_0^t that is computed at denoising timestep t as:

$$I_0^t = \mathcal{D}((x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta)/\sqrt{\bar{\alpha}_t}), \tag{5}$$

where $\bar{\alpha}_t$ is the coefficient of the DDPM noise scheduler [21], ϵ_θ is the estimated noise, and $\mathcal{D}$ is the VAE decoder. The total training objective becomes:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{noise}} + \lambda_2 \ \text{LPIPS}\left(I_0^t, I_i\right) \tag{6}$$

where LPIPS is the model used to compute LPIPS loss, and λ_1 and λ_2 are the loss weighing hyperparameters.

During inference, to apply an edit to a new query image I_q, the model is conditioned on the EditCLIP embedding produced from the exemplar image pair I_i and I_e, while the latent representation of the query image is concatenated with the noise x_t. This effectively modifies Equation (1) to perform exemplar-based image editing, generating an output image I_o, which is the edited version of I_q:

$$I_o = \mathcal{U}(I_q; \tilde{\mathcal{F}}_\theta(\texttt{concat}(I_i, I_e))) \ , \tag{7}$$

where $\mathcal{U}$ is our editing model.

3.4. EditCLIP for Evaluating Edits

As demonstrated in the Figure 3, EditCLIP effectively captures semantic changes between the reference image and its edited counterpart. At the same time, the EditCLIP embeddings E are trained to exhibit high similarity with the textual embedding $\mathcal{T}$ of the respective editing instruction T. Leveraging these properties, we can assess how well a performed edit aligns with a given editing textual instruction.

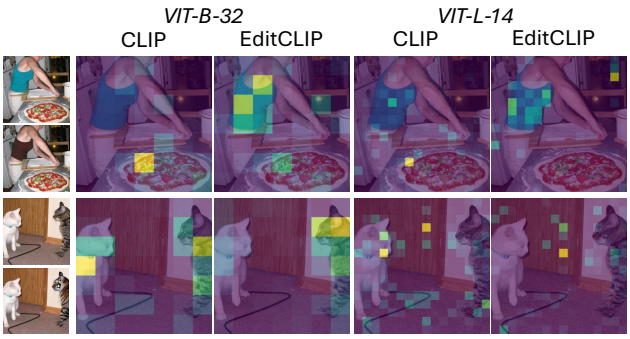

Figure 3. A visualization of the visual encoder's attention in EditCLIP compared to the original CLIP. We visualize the attention of the $[CLS]$ token from the last attention head. Unlike CLIP, where attention is dispersed across the image, EditCLIP focuses on the differences between the input and edited image, indicating that it effectively captures the edited regions.

Given an input image I_i, an arbitrary editing approach generates an edited image I_e based on the editing instruction T. We define the EditCLIP-to-Text (**EC2T**) similarity metric as:

$$\textbf{EC2T}(I_i, I_e, T) = \texttt{cos}(\tilde{\mathcal{F}}_\theta(\texttt{concat}(I_i, I_e)), T) \ , \quad (8)$$

where cos denotes cosine similarity. This metric quantifies how the input image transforms into the edited image and whether the changes align with the specified editing instructions. Unlike existing metrics based on the original CLIP, our edit embeddings implicitly capture all changes between the reference and edited images. This enables the evaluation of complex edits while penalizing undesired changes in the image that were not specified in the editing instructions.

In exemplar-based image editing, where both the reference input image I_i and its edited counterpart I_e are provided, the goal is to apply the same edit to a query image I_q without requiring textual instruction. Given an output image I_o produced by an arbitrary exemplar-based editing approach, another metric, EditCLIP-to-EditCLIP (**EC2EC**), can be computed as:

$$\textbf{EC2EC}(I_i, I_e, I_q, I_o) = \texttt{cos}(\tilde{\mathcal{F}}_\theta(\texttt{concat}(I_i, I_e)) \; \tilde{\mathcal{F}}_\theta(\texttt{concat}(I_q, I_o))) \ . \quad (9)$$

This metric would capture how similar the edit is between the reference and the target pairs.

4. Experiments

We demonstrate the effectiveness of our EditCLIP model on two tasks: (1) exemplar-based image editing and (2) automated evaluation of image editing methods. To ensure reliable evaluation, we complement our experiments with user studies conducted by humans to validate our findings.

4.1. Experimental Setup

Training Dataset: We employ the Instruct-Pix2Pix (IP2P) [4] image editing dataset for both EditCLIP pre-training and for exemplar-based image editing. The dataset is instruction-based and contains around 313k filtered input/edit/instruction triplets [1]. The edit types in the dataset primarily consist of global style transfer and local object addition or replacement.

EditCLIP Pre-training: We initialize our model from pre-trained CLIP models [33], modifying and fine-tuning the visual encoder as explained in Section 3.2 while keeping the text encoder frozen. We apply a learning rate of $2e-4$ to the first convolution layer, which processes both the reference input and edited image, and we use a lower learning rate of $2e-6$ for all other layers. We experiment with two CLIP variations that are commonly used, `ViT-B/32` and `ViT-L/14`. Each model is trained until convergence, with the former converging after 35 epochs and the latter after 40 epochs. All training was conducted on 4 NVIDIA A100-80G GPUs with a per-GPU batch size of 256.

Exemplar-based Editing Training and Inference: We adopt the base training setup from IP2P, using Stable Diffusion 1.5 [35] as the base model, and initialize it with the weights from the pre-trained IP2P. For the loss in Equation (6), we set the weights $\lambda_1 = 1$ and $\lambda_2 = 0.05$. We use a constant learning rate of $5e-5$ throughout the training and train for $16k$ iterations. The training was done on a single NVIDIA A100-80G with a batch size of 64. During inference, we use a fixed edit guidance scale $s_E = 7$ for edit embedding E and image guidance scale $s_I = 1.5$ for input image I_i (see the supplementary materials for more details).

Evaluation Benchmark: We adapt the TOP-Bench dataset [53] for exemplar-based image editing and we denote it as *TOP-Bench-X*. TOP-Bench consists of different types of edits, where each type includes a set of training and test pairs. We use the training set to form exemplar pairs, denoted as $[I_i, I_e]$, while the test set provides the corresponding query image I_q. This results in a total of 1277 samples, comprising 257 unique exemplars and 124 unique queries. We employ this benchmark to evaluate both exemplar-based image editing and the alignment of our proposed metrics with human judgment. To assess the perceptual quality of edits, we conducted a two-alternative forced-choice (2AFC) user study on Amazon Mechanical Turk. Participants rated two criteria: (1) the quality of the edits and (2) the preservation of query image details (see supplementary materials for further details).

4.2. Exemplar-based Image Editing

Here, we demonstrate the capabilities of our proposed EditCLIP embeddings as an editing conditioning signal, re-

[1] https://huggingface.co/datasets/timbrooks/instructpix2pix-clip-filtered

		Text-based			*Exemplar-based*		*User-Study*		
	LPIPS ↓	CLIP ↑	**EC2T** ↑	CLIP-Dir. ↑	S_{visual}[32] ↓	**EC2EC** ↑	WR-Edit ↑	WR-Pres ↑	RT (s)
IP2P[4]*	0.295	0.226	0.196	0.232	0.710	0.418	55.64	52.31	1.8
VISII[32]	0.518	0.203	0.152	0.096	0.832	0.313	79.84	79.43	370
PromptDiffusion[43]	0.620	0.180	0.127	0.041	0.906	0.204	91.80	89.65	5.5
InstaManip[29]	0.383	**0.226**	0.168	**0.161**	**0.735**	0.383	51.21	53.43	14.9
EditCLIP (Ours)	**0.233**	0.216	**0.180**	0.143	0.761	**0.477**	-	-	**1.8**

Table 1. Quantitative results for exemplar-based image editing. *IP2P is text-based, but we include it as a reference. WR-Edit and WR-Pres denote the winning rate of edit quality and input preservation of *our method against other methods* according to human evaluators. RT refers to runtime in seconds. We show the best one in **bold font** and the second best in underline. Our approach performs on par with the recent SOTA method, InstaManip, despite having only 20 times fewer parameters.

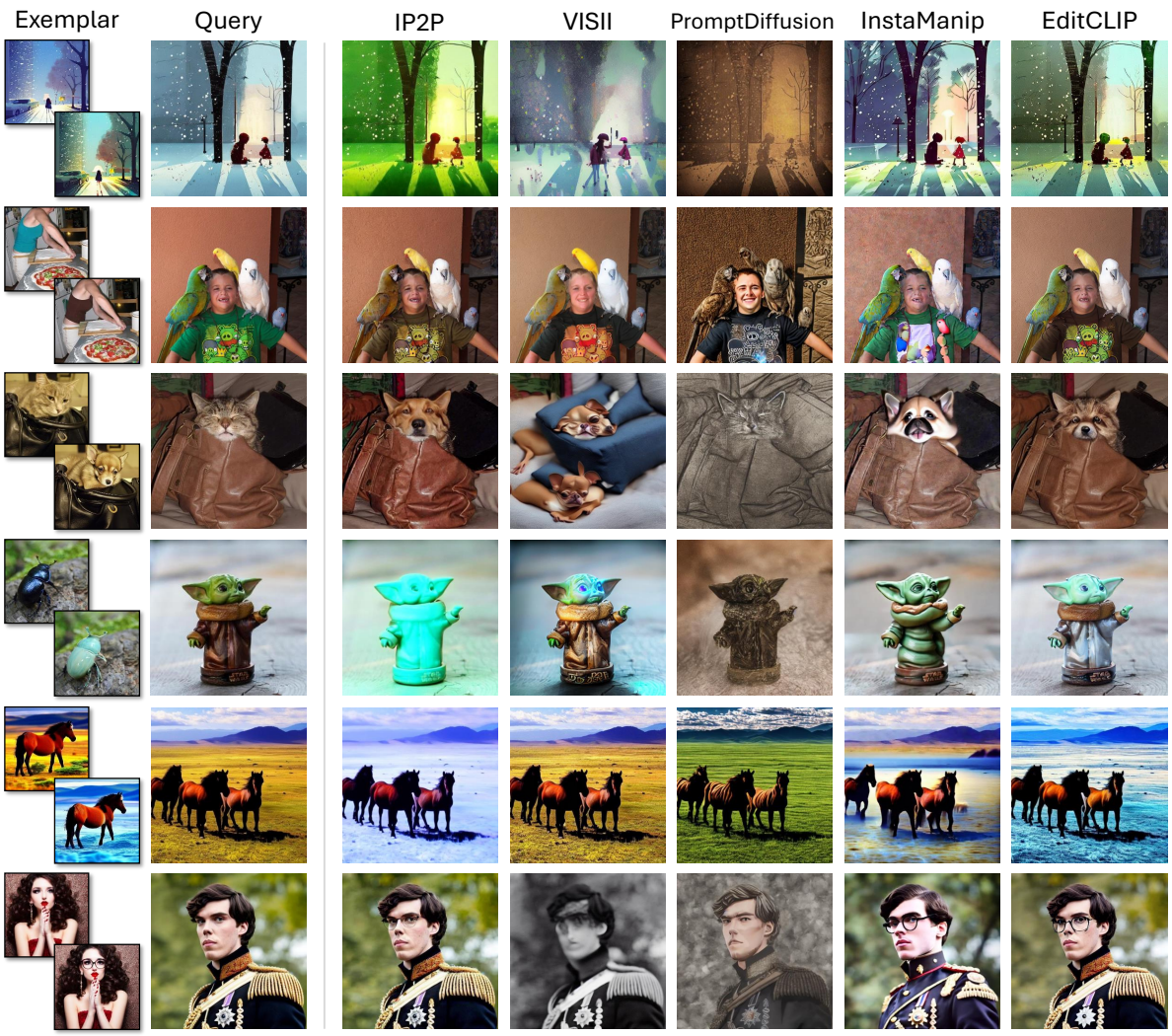

Figure 4. Qualitative comparison for exemplar-based image editing.

placing text-based instructions in image editing.

Baselines: We compare against existing exemplar-based approaches with publicly available source code, including VISII [32], PromptDiffusion (PD) [43], and the recent InstaManip [29]. Additionally, we include IP2P [4] as a reference for how an instruction-based approach would perform

Figure 5. EditCLIP can perform complex edits when the exemplars contain multiple edits in a single step.

in comparison. For all methods in comparison, we follow the original setups of their respective code bases. For improved credibility, we run each evaluation sample with 5 different random seeds for every method.

Quantitative Results: We evaluate on *TOP-Bench-X* and report the results in Section 4. We include the *exemplar-based* metric S_{visual} [32], along with our proposed **EC2EC** metric, described in Section 3.4, and a user study to validate our findings. Our approach performs the best on **EC2EC**, a result that is confirmed by the user study with our winning rate larger than 50% against all baselines, demonstrating superior edit quality and better preservation of the query image structure. We also include the commonly used LPIPS and text-based metrics, including CLIP Score, CLIP Directional Similarity, and our proposed **EC2T** metric for completeness. Note that these metrics are computed using the textual editing instruction T provided by the benchmark or the textual description of the output image. As expected, IP2P achieves the best performance on all text-based metrics, as it employs the textual instruction as a conditioning signal. Additionally, as our model is finetuned on text-based IP2P, it is challenging to bridge the modality gap between original text conditioning and EditCLIP conditioning, especially when we only finetune on $313k$ samples. However, our approach performs the best on **EC2T** among exemplar-based approaches, which aligns with the user study. In terms of runtime, our method is the fastest, as it neither requires test-time optimization like VISII nor employs large Vision-Language Models (VLMs) as in InstaManip. More details on the metrics and the user study can be found in the supplementary materials.

Qualitative Results: To facilitate qualitative comparisons with the baselines, we evaluate on selected samples in prior work [7, 15, 32, 38, 42, 50]. Figure 4 shows the qualitative comparison between our method using EditCLIP with the `VIT-L-14` backbone. We provide the results obtained by the backbone `VIT-B-32` in the supplementary materials. Our approach excels across various types of edits, including global style transfer, color modification, object addition and swapping, and material editing. IP2P performs well at edits that are easily described in text, *e.g.*, "adding glasses" or "changing a cat to a dog," but struggles with edits such as style or material transfer, as these edits are often difficult to express in text. This highlights the effectiveness of our EditCLIP embeddings in capturing edits that are not easily described through text.

VISII performs reasonably on style transfer but struggles with other types of edits, as its test-time optimization may diverge. The recent state-of-the-art method, InstaManip, demonstrates strong performance across various types of edits; however, this comes at a significant computational cost due to its reliance on a huge VLM. In contrast, our method outperforms InstaManip in accurately applying fine details with high fidelity while preserving the original layout, all at a drastically lower computational cost.

Multi-Edit Examples: To demonstrate the effectiveness of EditCLIP in handling complex exemplars with multiple edits, we present challenging editing cases where multiple edits are present in the exemplar. For iP2P, we construct two different variations: *IP2P (combined)*, which receives a single textual instruction combining all edits and performs them in one step and *IP2P (multi-turn)*, which receives separate textual instructions for each edit and applies them sequentially over multiple steps. For both InstaManip and our method, all edits are performed in a single shot. As shown in Figure 5, our method successfully transfers multiple edits from the exemplar in just one shot, while both IP2P and InstaManip fail.

Ablation Study: To demonstrate the effectiveness of EditCLIP embeddings over the original CLIP, we experiment with different conditioning setups for capturing the edits using the original CLIP only. We only modify the conditioning embedding E to reflect these changes, but we keep all training and inference parameters the same. The setups that we explore are:

- $\mathcal{F}_\theta(I_e)$: Embedding of the reference edit image.
- $\mathcal{F}_\theta(I_i) + \mathcal{F}_\theta(I_e)$: Sum of the reference input and edited image embeddings.
- $\mathcal{F}_\theta(I_i) \otimes \mathcal{F}_\theta(I_e)$: Concatenation of the reference input and edited image embeddings along the channel dim.

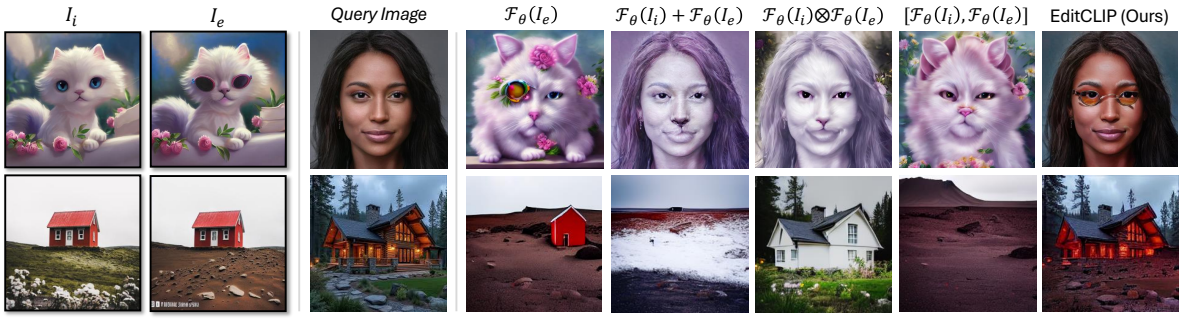

Figure 6. Ablation of different conditioning embeddings using the original CLIP visual encoder $\mathcal{F}_\theta$. EditCLIP embedding greatly outperforms all CLIP variations.

- $[\mathcal{F}_\theta(I_i), \mathcal{F}_\theta(I_e)]$: Appending the reference input and edited image embeddings along the sequence length dim.

All these embeddings are extracted from the original CLIP. We present the comparison in Figure 6. $\mathcal{F}_\theta(I_e)$ fails to capture the edit, causing the model to generate only variations of the reference input image. For all other variations that utilize both the reference input and edit images, the model struggles to identify the intended edit and instead blends the two images uncontrollably. In contrast, our EditCLIP embeddings effectively capture the edits and accurately transfer them to the query image without altering its structure. This advocates for the embedding produced by EditCLIP over a combined embedding approach using the original CLIP. Notably, in the first row, the reference exemplar exhibits a slight global style change, making the image more saturated. EditCLIP accurately captures this adjustment alongside the intended edit of adding sunglasses. We provide additional ablations in the supplementary.

4.3. Automated Evaluation of Image Editing

To evaluate how well existing CLIP-based metrics, including CLIP, Directional CLIP, and S_{visual}, as well as our proposed metrics **EC2EC** and **EC2T** in Section 3.4, align with human evaluation, we compute the Pearson correlation between human judgments and each of these metrics in Section 4.3. To evaluate *text-based* metrics for instruction-based editing, we evaluate edited samples on TOP-Bench-X benchmark generated using IP2P and two additional approaches: Ledits++ [3] and EF-DDPM [23]. For the text-based metrics, our proposed **EC2T** achieves the highest correlation with human judgment both in edit quality and image preservation, indicating a better alignment with humans. For evaluating *exemplar-based* metrics, we referred to the same user study mentioned in Sec. 4.2. Our **EC2EC** achieves a higher correlation than S_{visual} both in edit quality and image preservation. These results showcase that our proposed EditCLIP embeddings are more reliable metrics for automated evaluation of both instruction-based and exemplar-based image editing methods. Although **EC2EC** and **EC2T** are trained using the IP2P dataset as their backbone, their performance is evaluated exclusively on the TOP-Bench-X benchmark to prevent self-evaluation bias.

	Text-based			*Exemplar-based*	
	CLIP	CLIP-Dir.	**EC2T**	S_{visual}	**EC2EC**
Edit	0.209	0.186	**0.256**	0.240	**0.372**
Preserves	-0.028	-0.023	**0.104**	-0.023	**0.157**

Table 2. Pearson correlation between individual metrics and human judgment in terms of edit quality and input preservation. Our proposed metrics achieve the highest correlation with human evaluation demonstrating better alignment.

5. Limitations and Future Work

EditCLIP is trained solely on the IP2P dataset [4], which lacks edits like removal and deformation. Expanding training data with additional datasets could improve the quality and diversity of the editing embedding space. Please refer to the supplementary material for examples of failure cases.

For future work, EditCLIP could be applied to downstream tasks like instruction caption generation, query-based editing pair retrieval, and extensions to video and 3D editing. Further improvements include exploring advanced training strategies, such as refined loss functions [49] or augmented text instructions [13], and incorporating masks as an extra channel to enhance control over edit regions.

6. Conclusion

We proposed EditCLIP, a representation-learning approach for image editing that captures how images transform during edits. Experiments showed that EditCLIP achieves state-of-the-art exemplar-based image editing with no computational overhead. Moreover, we showed that EditCLIP serves as a reliable metric for evaluating edit quality and faithfulness to the reference image, aligning closely with human judgment. Such a metric can accelerate the development of image editing approaches by providing an evaluation metric that aligns better with human judgment compared to existing metrics.

Acknowledgements

This work was supported by funding from King Abdullah University of Science and Technology (KAUST) - Center of Excellence for Generative AI, under award number 5940.

References

[1] Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. Cross-image attention for zero-shot appearance transfer. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–12, 2024. 2

[2] Manuel Brack, Felix Friedrich, Dominik Hintersdorf, Lukas Struppek, Patrick Schramowski, and Kristian Kersting. Sega: Instructing diffusion using semantic dimensions. *NeurIPS*, 2023. 2

[3] Manuel Brack, Felix Friedrich, Katharina Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinaros Passos. Ledits++: Limitless image editing using text-to-image models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2, 8

[4] 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*, pages 18392–18402, 2023. 1, 2, 3, 4, 5, 6, 8

[5] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22560–22570, 2023. 1, 2

[6] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 397–406, 2021. 4

[7] Ta-Ying Cheng, Prafull Sharma, Andrew Markham, Niki Trigoni, and Varun Jampani. Zest: Zero-shot material transfer from a single image. *ECCV*, 2024. 7

[8] Jooyoung Choi, Yunjey Choi, Yunji Kim, Junho Kim, and Sungroh Yoon. Custom-edit: Text-guided image editing with customized diffusion models. *arXiv preprint arXiv:2305.15779*, 2023. 2

[9] Jiwoo Chung, Sangeek Hyun, and Jae-Pil Heo. Style injection in diffusion: A training-free approach for adapting large-scale diffusion models for style transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8795–8805, 2024. 2

[10] Aleksandar Cvejic, Abdelrahman Eldesokey, and Peter Wonka. Partedit: Fine-grained image editing using pre-trained diffusion models. *arXiv preprint arXiv:2502.04050*, 2025. 2

[11] Wenkai Dong, Song Xue, Xiaoyue Duan, and Shumin Han. Prompt tuning inversion for text-driven image editing using diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7430–7440, 2023. 2

[12] Patrick Esser, Sumith Kulal, A. Blattmann, Rahim Entezari, Jonas Muller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. *ArXiv*, 2024. 1

[13] Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving clip training with language rewrites. In *Advances in Neural Information Processing Systems*, pages 35544–35575. Curran Associates, Inc., 2023. 8

[14] Yuying Ge, Yixiao Ge, Ziyun Zeng, Xintao Wang, and Ying Shan. Planting a seed of vision in large language model. *arXiv preprint arXiv:2307.08041*, 2023. 2

[15] Yuying Ge, Sijie Zhao, Chen Li, Yixiao Ge, and Ying Shan. Seed-data-edit technical report: A hybrid dataset for instructional image editing, 2024. 7

[16] Yuying Ge, Sijie Zhao, Ziyun Zeng, Yixiao Ge, Chen Li, Xintao Wang, and Ying Shan. Making llama see and draw with seed tokenizer. *ICLR*, 2024. 2

[17] Daniel Geng and Andrew Owens. Motion guidance: Diffusion-based image editing with differentiable motion estimators. *International Conference on Learning Representations*, 2024. 1

[18] Qin Guo and Tianwei Lin. Focus on your instruction: Fine-grained and multi-instruction image editing by attention modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6986–6996, 2024. 2

[19] Ligong Han, Song Wen, Qi Chen, Zhixing Zhang, Kunpeng Song, Mengwei Ren, Ruijiang Gao, Anastasis Stathopoulos, Xiaoxiao He, Yuxiao Chen, et al. Improving tuning-free real image editing with proximal guidance. *arXiv preprint arXiv:2306.05414*, 2023. 2

[20] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 1, 2

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 4

[22] Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8362–8371, 2024. 2

[23] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12469–12478, 2024. 1, 2, 8

[24] 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 *Pro-*

ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6007–6017, 2023. 2

[25] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2426–2435, 2022. 2, 3

[26] Jing Yu Koh, Daniel Fried, and Ruslan Salakhutdinov. Generating images with multimodal language models. *NeurIPS*, 2023. 2

[27] Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. *arXiv preprint arXiv:2412.08629*, 2024. 1

[28] Black Forest Labs. Flux.1 [dev], 2024. Accessed: 2025-11-14. 1

[29] Bolin Lai, Felix Juefei-Xu, Miao Liu, Xiaoliang Dai, Nikhil Mehta, Chenguang Zhu, Zeyi Huang, James M Rehg, Sangmin Lee, Ning Zhang, and Tong Xiao. Unleashing in-context learning of autoregressive models for few-shot image manipulation. *arXiv preprint arXiv:2412.01027*, 2024. 2, 3, 6

[30] Bingyan Liu, Chengyu Wang, Tingfeng Cao, Kui Jia, and Jun Huang. Towards understanding cross and self-attention in stable diffusion for text-guided image editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7817–7826, 2024. 2

[31] Hyelin Nam, Gihyun Kwon, Geon Yeong Park, and Jong Chul Ye. Contrastive denoising score for text-guided latent diffusion image editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9192–9201, 2024. 2

[32] Thao Nguyen, Yuheng Li, Utkarsh Ojha, and Yong Jae Lee. Visual instruction inversion: Image editing via image prompting. *Advances in Neural Information Processing Systems*, 36:9598–9613, 2023. 2, 3, 6, 7

[33] 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*, pages 8748–8763. PMLR, 2021. 2, 3, 4, 5

[34] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 1

[35] 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*, pages 10684–10695, 2022. 5

[36] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1

[37] Yujun Shi, Chuhui Xue, Jiachun Pan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. *CVPR*, 2024. 1

[38] Ashutosh Srivastava, Tarun Ram Menta, Abhinav Java, Avadhoot Jadhav, Silky Singh, Surgan Jandial, and Balaji Krishnamurthy. Reedit: Multimodal exemplar-based image editing with diffusion models. *arXiv preprint arXiv:2411.03982*, 2024. 2, 3, 7

[39] 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)*, pages 1921–1930, 2023. 2

[40] Dani Valevski, Matan Kalman, Eyal Molad, Eyal Segalis, Yossi Matias, and Yaniv Leviathan. Unitune: Text-driven image editing by fine tuning a diffusion model on a single image. *ACM Trans. Graph.*, 42(4), 2023. 2

[41] Kai Wang, Fei Yang, Shiqi Yang, Muhammad Atif Butt, and Joost van de Weijer. Dynamic prompt learning: Addressing cross-attention leakage for text-based image editing. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[42] Qian Wang, Biao Zhang, Michael Birsak, and Peter Wonka. MDP: A generalized framework for text-guided image editing by manipulating the diffusion path. *Transactions on Machine Learning Research*, 2024. 7

[43] Zhendong Wang, Yifan Jiang, Yadong Lu, yelong shen, Pengcheng He, Weizhu Chen, Zhangyang "Atlas" Wang, and Mingyuan Zhou. In-context learning unlocked for diffusion models. In *Advances in Neural Information Processing Systems*, pages 8542–8562. Curran Associates, Inc., 2023. 2, 6

[44] Chen Henry Wu and Fernando De la Torre. A latent space of stochastic diffusion models for zero-shot image editing and guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7378–7387, 2023. 2

[45] Zongze Wu, Nicholas Kolkin, Jonathan Brandt, Richard Zhang, and Eli Shechtman. Turboedit: Instant text-based image editing. *ECCV*, 2024. 1

[46] Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with language-guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9452–9461, 2024. 1, 2

[47] Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with natural language. 2024. 2

[48] Yifan Yang, Houwen Peng, Yifei Shen, Yuqing Yang, Han Hu, Lili Qiu, Hideki Koike, et al. Imagebrush: Learning visual in-context instructions for exemplar-based image manipulation. *Advances in Neural Information Processing Systems*, 36:48723–48743, 2023. 2, 3

[49] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training . In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Los Alamitos, CA, USA, 2023. IEEE Computer Society. 8

[50] 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*, 36:31428–31449, 2023. 2, 7

[51] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 3

[52] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 4

[53] Ruoyu Zhao, Qingnan Fan, Fei Kou, Shuai Qin, Hong Gu, Wei Wu, Pengcheng Xu, Mingrui Zhu, Nannan Wang, and Xinbo Gao. Instructbrush: Learning attention-based instruction optimization for image editing. *arXiv preprint arXiv:2403.18660*, 2024. 2, 3, 5