LIME: Localized Image Editing via Attention Regularization in Diffusion Models

Supplementary Material

We further explore LIME's applicability to other models like Prompt-to-Prompt [18], HIVE [54], and InstructDiffusion [16], showing improved localized editing through quantitative and qualitative results across datasets like MagicBrush [53], PIE-Bench [21], and EdilVal [5]. We provide ablation studies and implementation details and discuss broader impacts like potential misuse risks balanced against benefits like enhanced creative expression. The material underscores LIME's ability to enable precise localized image edits while preserving surrounding context.

A. Additional Experiments

A.1. Applicability of LIME to other models

The core concepts behind LIME make it broadly applicable to a variety of image editing models, including Prompt-to-Prompt [18], HIVE [54], and InstructDiffusion [16]. Integrating LIME into these methods offers the potential for enhanced performance in the following ways:

Prompt-to-Prompt is a prompt-based editing method [18]. Unlike instruction-based editing techniques such as IP2P, an input caption and an output caption are needed to execute the desired image edit. Moreover, since there is no condition on the input image, an image inversion step is required before applying the edit. Prompt-to-Prompt does offer localized editing capabilities, with an extension of the *Blend* option, which mixes the diffusion processes of two images (original and edited). However, as seen in Fig. 8, integrating the edit application of LIME into Prompt-to-Prompt enables even more precise localized edits. This component, see Sec. 4.2, ensures that changes are confined to the RoI while preserving the shapes and context of surrounding elements. The official code base[1] is used for comparison.

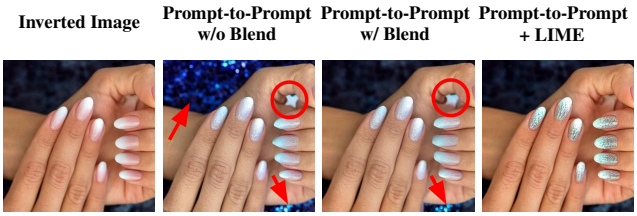

Figure 8. **The integration of LIME into Prompt-to-Prompt [18].** Red circles and arrows emphasize the localization issues of Prompt-to-Prompt model. The following textual information is used: **input caption:** *A photo of fingernails* and **output caption:** *A photo of glitter fingernails*.

HIVE is a fine-tuned version of IP2P on an expanded dataset. Further refinement is achieved through fine-tuning with a reward model, which is developed based on human-ranked data. Table 3 shows the results on the MagicBrush dataset. HIVE [54] improves the performance of IP2P (compare the first and third rows), and further improvement is achieved by fine-tuning HIVE on MagicBrush training set, MB with ✓ stands for it in Tab. 3, (compare the third and fifth rows). For both base HIVE and the version fine-tuned on MB, LIME can further significantly improve performance (compare the third and fourth rows and fifth and sixth rows.).

Fig. 9 presents a qualitative comparison before and after integrating LIME into the HIVE model on samples from the MagicBrush dataset. Figure 9-(a) displays the color change of an object in a scene. While HIVE can effectively implement the edit, it inadvertently alters the structure of the vase and the color of another vase in the background, as highlighted by the red circle. However, with our model integrated, HIVE accurately targets the plants for the desired edit without affecting unrelated areas. In Fig. 9-(b), HIVE fails to recognize one of the fingernails and does not correctly apply the intended edit, indicated by the red circle and arrows. Additionally, HIVE alters the background color to blue, which was mentioned in the edit prompt. In contrast, HIVE, with our model integration, precisely applies edits to the region of interest, such as the fingernails, without missing any parts or altering areas outside the region of interest. Lastly, Fig. 9-(c) demonstrates another instance where HIVE performs an entangled edit, changing the skin color of a

[1] *https://github.com/google/prompt-to-prompt/*

Table 3. **HIVE + LIME Evaluation on MagicBrush [53].** The numbers for others are sourced from [53], while values for our method are computed by following the same protocol.The integration of **LIME** surpasses the base model performance, *e.g.*, HIVE and HIVE w/MB.

Methods	MB	Single-turn					Multi-turn				
		L1 ↓	L2 ↓	CLIP-I ↑	DINO ↑	CLIP-T ↑	L1 ↓	L2 ↓	CLIP-I ↑	DINO ↑	CLIP-T ↑
IP2P [7]	✗	0.112	0.037	0.852	0.743	0.276	0.158	0.060	0.792	0.618	0.273
IP2P [7]	✓	0.063	0.020	0.933	0.899	0.278	0.096	0.035	0.892	0.827	0.275
HIVE [54]	✗	0.109	0.034	0.852	0.750	0.275	0.152	0.056	0.800	0.646	0.267
HIVE [54] + **LIME**	✗	**0.051**	**0.016**	**0.940**	**0.909**	0.293	**0.080**	0.029	0.894	**0.829**	0.283
HIVE [54]	✓	0.066	0.022	0.919	0.866	0.281	0.097	0.037	0.879	0.789	0.280
HIVE [54] + **LIME**	✓	0.053	**0.016**	0.939	0.906	**0.300**	**0.080**	**0.028**	**0.899**	**0.829**	**0.295**

woman in the scene when the intended edit was to change the outfit color, as shown by the red arrow. The integration of our model enables localized and separate edits on the input image based on the edit instructions, ensuring that only the specified changes are made.

Figure 9. **The integration of LIME into HIVE [54].** Red circles and arrows emphasize the localization issues of HIVE model. HIVE + LIME enables localized and effective edits.

InstructionDiffusion (IDiff) is another IP2P-based method [16]. Integrating LIME into IDiff enhances its performance. IDiff achieved scores of 0.085, 0.03, 0.90, 0.83, and 0.30 while the integration of LIME into IDiff improves the scores to 0.071, 0.02, 0.92, 0.86, and 0.30 for L1, L2, CLIP-I, DINO, and CLIP-T, respectively, on MagicBrush test dataset. The results can also be comparable with Tab. 3.

Instruction: *Replace the surfboards with flowers.*

A.2. MagicBrush Mask Annotations

As mentioned in Sec. 5.5, the mask annotations for the MagicBrush dataset [53] are not very tight around the edit area which might result in worse edit quality when we use them rather than the segmentation extracted by LIME. We show qualitative results highlighting the problem in Fig. 10. Our method directly uses the identified mask during the editing process, therefore, it is important for the masks to be as tight as possible around the edit area to apply localized edits. The loose GT masks of MagicBrush explain why our model achieves worse performance in Tab. 2 when using GT masks. We highlight the significance of precise masks with red circles in Fig. 10. When precise masks are provided to LIME, localized edits can be achieved. For the first row - (a), the handle of the racket can be preserved if the mask has a precise boundary between the handle and outfit in the occluded area. Moreover, the second row - (b) shows that if the mask in the MagicBrush dataset is used during the edit, the method changes the color of the blanket as well. However, with the precise mask extracted by our method, the edit can distinguish the objects in the area and apply localized edits.

These results highlight the quality of the edit masks extracted by our method in an entirely self-supervised way and hint at possible further developments where our contribution could be used to refine the annotations of existing datasets or speed up the creation of new ones.

Table 4. **Evaluation on PIE-Bench [21].** Comparison across ten edit types shows the integration of **LIME** outperforming base models on instruction-based editing models. *GT Mask* stands for ground-truth regions of interest masks.

	Structure	Background Preservation				CLIP Similarity	
Methods	**Distance**$_{\times 10^3}$ ↓	**PSNR** ↑	**LPIPS**$_{\times 10^3}$ ↓	**MSE**$_{\times 10^4}$ ↓	**SSIM**$_{\times 10^2}$ ↑	**Whole** ↑	**Edited** ↑
InstructDiffusion [16]	75.44	20.28	155.66	349.66	75.53	23.26	21.34
DirectInversion + P2P [21]	11.65	27.22	54.55	32.86	84.76	**25.02**	**22.10**
IP2P [7]	57.91	20.82	158.63	227.78	76.26	23.61	21.64
IP2P [7] **+ LIME**	32.80	21.36	110.69	159.93	80.20	23.73	21.11
IP2P w/MB [53]	22.25	27.68	47.61	40.03	85.82	23.83	21.26
IP2P w/MB [53] **+ LIME**	**10.81**	**28.80**	**41.08**	**27.80**	**86.51**	23.54	20.90
HIVE [54]	56.37	21.76	142.97	159.10	76.73	23.30	21.52
HIVE [54] **+ LIME**	37.05	22.90	112.99	107.17	78.67	23.41	21.12
HIVE w/MB [53]	34.91	20.85	158.12	227.18	76.47	23.90	21.75
HIVE w/MB [53] **+ LIME**	26.98	26.09	68.28	63.70	84.58	23.96	21.36

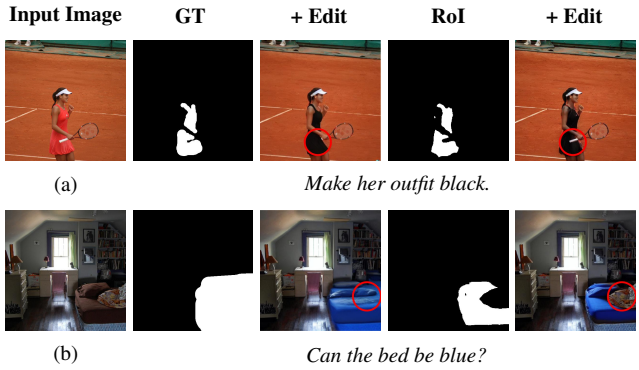

Figure 10. **MagicBrush Mask Annotations.** Ground truth (GT) refers to mask annotations in MagicBrush [53]. RoI indicates inferred masks from our proposed method. Red circles on the edited images (+ Edit) highlight area where the precise localization of the edits can be appreciated.

A.3. More Quantitative Results

PIE-Bench [21] The benchmark includes 700 images in 10 editing categories with input/output captions, editing instructions, input images, and RoI annotations. Metrics for *structural integrity* and *background preservation* are derived from cosine similarity measures and image metrics like *PSNR*, *LPIPS*, *MSE*, and *SSIM*, while text-image consistency is evaluated via *CLIP Similarity*.

Quantitative analysis on PIE-Bench [21] demonstrates the effectiveness of our proposed method. Compared to baseline models like IP2P [7] and the fine-tuned version on MagicBrush [53] and HIVE [54], our method achieves significantly better performance on metrics measuring structure and background preservation. This indicates that our approach makes localized edits according to the instructions while avoiding unintended changes to unaffected regions. *Edited* measures CLIP similarity between edit prompt and edited area, and with *Background Preservation* provides a measure of localized edits. As seen in Tab. 4, our method is better if both metrics are considered together. At the same time, our method obtains comparable results to base models on the CLIP similarity score, showing that edits are faithfully applied based on the textual instruction. A comprehensive comparison is presented in Tab. 4. Overall, the quantitative results validate that our method can enable text-guided image editing by making precise edits solely based on the given edit instruction without altering unrelated parts.

EditVal [5] The benchmark offers 648 image editing operations spanning 19 classes from the MS-COCO dataset [26]. The benchmark assesses the success of each edit with a binary score that indicates whether the edit type was successfully applied. The OwL-ViT [31] model is utilized to detect the object of interest, and detection is used to assess the correctness of the modifications.

Table 5. **Evaluation on EditVal [5].** Comparison across six edit types shows our method outperforming eight state-of-the-art text-guided image editing models. The numbers for other methods are directly taken from the benchmark dataset [5].

Method	**O.A.**	**O.R.**	**P.R.**	**P.A.**	**S.**	**A.P.**	**Avg.**
SINE [55]	0.47	**0.59**	0.02	0.16	0.46	0.30	0.33
NText. [33]	0.35	0.48	0.00	0.20	**0.52**	**0.34**	0.32
IP2P [7]	0.38	0.39	0.07	0.25	0.51	0.25	0.31
Imagic [22]	0.36	0.49	0.03	0.08	0.49	0.21	0.28
SDEdit [29]	0.35	0.06	0.04	0.18	0.47	0.33	0.24
DBooth [42]	0.39	0.32	0.11	0.08	0.28	0.22	0.24
TInv. [15]	0.43	0.19	0.00	0.00	0.00	0.21	0.14
DiffEdit [9]	0.34	0.26	0.00	0.00	0.00	0.07	0.11
IP2P [7] **+ LIME**	**0.48**	0.49	**0.21**	**0.34**	0.49	0.28	**0.38**

Our method exhibits superior performance across various edit types in EditVal benchmark [5], particularly excelling in *Object Addition (O.A.)*, *Position Replacement (P.R.)*, and *Positional Addition (P.A.)*, while achieving second-best in *Object Replacement (O.R.)*. It performs on par with other methods for edits involving *Size (S.)* and *Alter Parts (A.P.)*. LIME advances the state-of-the-art by improving the average benchmark results by a margin of 5% over the previous best model, see Tab. 5.

A.4. Visual Comparison to state-of-the-art-methods

A.4.1 VQGAN-CLIP

As shown in Fig. 6a, VQGAN-CLIP [10] has better results on the *CLIP-T* metric. This is expected since it directly fine-tunes the edited images using CLIP embeddings. However, as seen in Fig. 11, the edited images from VQGAN-CLIP fail to preserve the details of the input image. On the other hand, our method successfully performs the desired edit by preserving the structure and fine details of the scene and results in a similar *CLIP-T* score as the one of the ground truth edited images in the MagicBrush dataset.

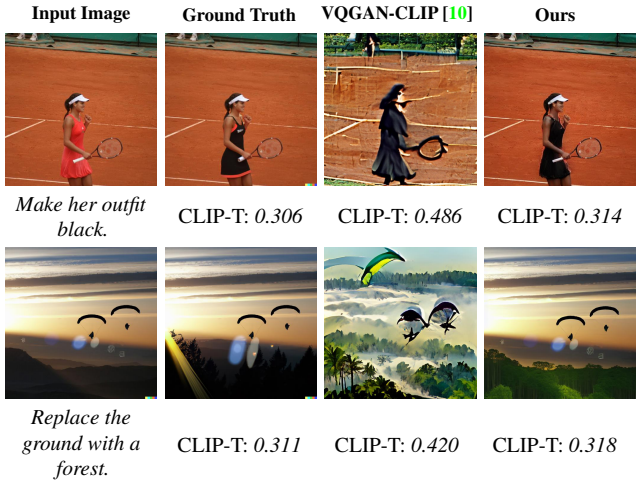

Figure 11. **Investigating CLIP-T for VQGAN-CLIP [10].** CLIP-T metrics are reported below each image and calculated between the output caption and the corresponding image. Input images and edit instructions are pictured in the first column. Ground truth edit images are taken from the MagicBrush dataset.

A.4.2 Diffusion Disentanglement

Wu *et al*. [50] propose a disentangled attribute editing method. Since it also claims disentangled (localized) edits, we visually compare our method + IP2P with Diffusion Disentanglement. This method is not disegned for instruction-based editing, so we use input and output captions during the comparison. Figure 12 shows edit types such as (a) texture editing and (b) replacing the object with a similar one. *Diffusion Disentanglement* on (a) alters the background objects in the image, *e.g*., adding snow on and changing the shape of the branch, and also changes the features of the object of interest, *e.g*., removing the tail of the bird. On (b), it fails to perform the desired edit altogether. Moreover, it requires a GPU with $>$ 48 GB RAM[2], and one image takes approximately 10 minutes on an NVIDIA A100 80GB to generate the edited version. In comparison, LIME achieves higher visual quality and takes 25 seconds to complete on NVIDIA A100 40GB with a GPU RAM usage of 25 GB.

[2] *https://github.com/UCSB-NLP-Chang/DiffusionDisentanglement*

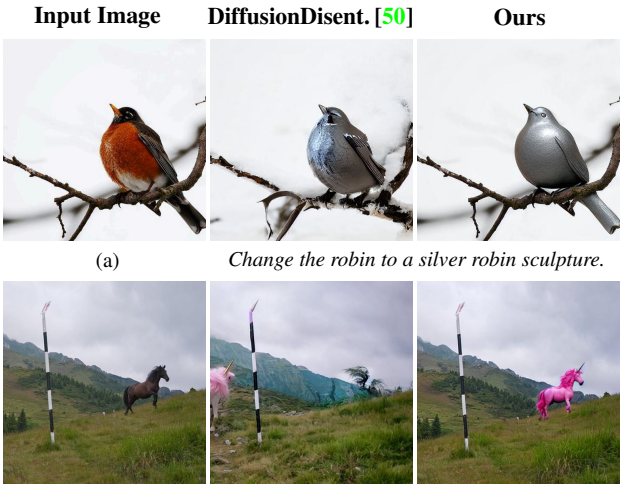

Figure 12. **Diffusion Disentanglement [50] Qualitative Comparison.** Edits are obtained by using the global description of the input image and the desired edit by concatenating them with ','.

A.4.3 Blended Latent Diffusion

As shown in Tab. 4, Blended Latent Diffusion [2] has better results than baselines and our method. However, as shown in Fig. 13, even if their method can perform the desired edit on the RoI from the user, (a) it distorts the location of the features, *e.g*., heads of the birds, and (b) it loses the information of the object in the input image and creates a new object in the RoI, *e.g*., blanket in (b). On the other hand, our method performs visually appealing edits on the input images considering the given edit instructions by preserving as many details from the input image as possible. This is also highlighted by a significantly lower Distance metric for our method in Tab. 4.

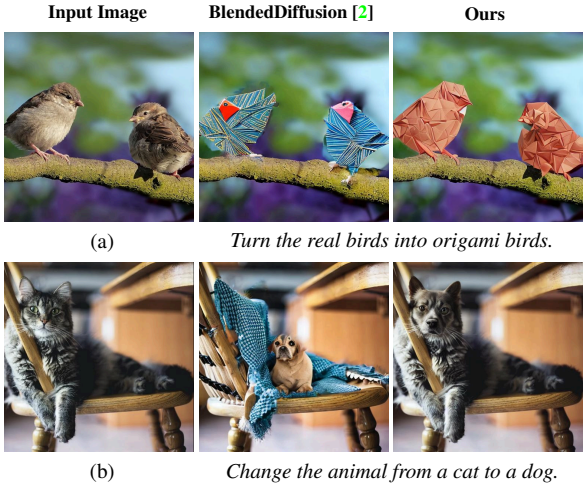

Figure 13. **BlendedDiffusion [2] Qualitative Comparison.** Edited images based on input images and edit instructions reported below each row. The images for BlendedDiffusion are taken from the PIE-Bench evaluation [21].

A.5. Qualitative comparison on segmentation maps

Our method proposes a segmentation method based on the clustering of intermediate features of the diffusion process. In this section, we provide a qualitative comparison to other segmentation methods that could be used as an alternative to this strategy. LPM [36] uses self-attention features from one resolution, such as 32×32, while our method leverages the intermediate features from different resolutions to enhance the segmentation map. Then, both apply a clustering method to find the segments in the input image. Another way to find segments is by using large segmentation models, *e.g.*, SAM [23], ODISE [51] .... As seen in Fig. 14 (i), large segmentation models cannot detect the transparent fin of the fish, while LPM and ours can. Moreover, LPM utilizes only one resolution, so it cannot find rocks in the river separately. As seen in Fig. 14 (ii), ODISE [51] and SAM [23] fail to segment minute object parts, like fingernails, while LPM and ours can find those segments. Furthermore, our method provides precise boundaries and segments in higher resolutions than LPM. Moreover, LPM uses Stable Diffusion [41] and requires real image inversion to find segments, while our method does not since it is based on IP2P [7]. For this reason, LPM requires more than 1 minute per image, while our proposal takes only 10-15 seconds per image. As a result, in a direct comparison to LPM, our method has the advantage of having higher-resolution segmentation maps with more details, and it is significantly faster. The publicly available official implementations of LPM[3], SAM[4] and ODISE[5] are used for the results in Fig. 14. Additionally, the same number of clusters is used for LPM and ours to achieve a fair comparison.

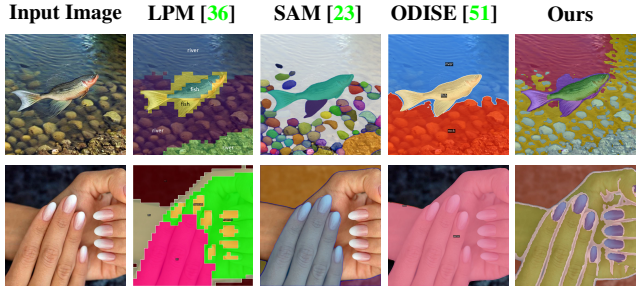

Figure 14. **Segmentation Qualitative.** Comparison between the-state-of-art segmentation methods on challenging examples. In the first example, both LPM and LIME can detect the fish fin, while SAM and ODISE cannot. Moreover, LIME provides a higher resolution than LPM. In the second example, SAM and ODISE fail to identify fingernails. Although LPM can find them, it delivers low-resolution and imprecise segments. Our method, however, accurately identifies fingernails and offers precise boundaries.

[3] *https://github.com/orpatashnik/local-prompt-mixing*
[4] *https://segment-anything.com/demo*
[5] *https://github.com/NVlabs/ODISE*

A.6. Comparison with open-vocabulary segmentors

An alternative to our proposed edit localization step, Sec. 4.1, would be to use off-the-shelves Open Vocabulary Segmentation models (OVS) and combine them with our *edit application*, see Sec. 4.2. A key difference between OVS and our localization method is that OVS requires an additional input, which is the *object of interest* to segment, and this could be a strong limitation in an instruction edit setting because the target is not always obvious from the instruction. As seen in Tab. 6, both, SEEM [56] and OV-SEG [25], significantly underperform compared to LIME.

Table 6. **OVS methods for localizing the edit.** We replace the edit localization part with OVS methods to show the significance of our proposed localization method. The methods are evaluated on MagicBrush dataset.

	Method	L1 ↓	L2 ↓	CLIP-I ↑	DINO ↑	CLIP-T ↑
	IP2P [7]	0.112	0.037	0.852	0.743	0.276
Mask Type	OV-SEG [25]	0.084	0.031	0.887	0.851	0.283
	SEEM [56]	0.079	0.022	0.903	0.866	0.289
	IP2P + LIME	**0.058**	**0.017**	**0.935**	**0.906**	**0.293**

In addition to quantitative analysis in Tab. 6, we provide a qualitative comparison for localizing the edit. As seen in Fig. 15, the OVS methods do not provide precise RoI for the edit instructions even when provided with a curated additional input, *e.g.*, object of interest. On the other hand, our proposed localization method in Sec. 4.1 provides precise and relevant RoIs without additional requirements.

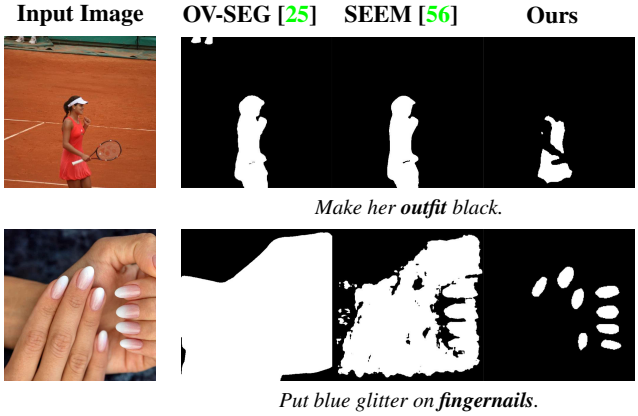

Figure 15. **OVS Method for Localization.** RoI examples that are found by different OVS methods, OV-SEG [25] and SEEM [56], and our proposal in Sec. 4.1. Since OVS methods require *object of interest*, words with **bold** are used for the *object of interest*.

A.7. Ablation study

A.7.1 Related Token Rewarding

In addition to the ablation study in Sec. 5.5, we also analyze token selection during cross-attention regularization as defined in Sec. 4.2. Instead of regularizing the attention of unrelated tokens, such as *<start of text>*, *padding*, and *stop words*, by penalizing it, we could think of doing the opposite and give high values to relevant tokens (denoted as $\tilde{S}$) within the RoI as reported in the following equation:

$$R(QK^T, M) = \begin{cases} QK^T_{ijt} + \infty, & \text{if } M_{ij} = 1 \text{ and } t \in \tilde{S} \\ QK^T_{ijt}, & \text{otherwise,} \end{cases} \tag{4}$$

This assignment guarantees that relevant tokens related to edit instructions have high scores after the softmax operation. As seen in Tab. 7, there is no significant improvement if the unrelated tokens are penalized instead of awarding the related tokens. However, penalizing the unrelated tokens gives the freedom to distribute the attention scores unequally among relevant tokens. Thus, it allows for a soft assignment of areas of the image among the related tokens.

Table 7. **Ablation Study on Token Selection.** For fair comparison, all parameters are the same for all settings except the ablated parameter.

Method	L1 ↓	L2 ↓	CLIP-I ↑	DINO ↑	CLIP-T ↑
IP2P [7]	0.112	0.037	0.852	0.743	0.276
Related	0.065	0.018	0.930	0.897	0.292
Unrelated	**0.058**	**0.017**	**0.935**	**0.906**	**0.293**

In addition to quantitative analysis on the MagicBrush dataset, Fig. 16 shows the attention scores for rewarding related tokens vs regularizing unrelated tokens. Even though the final edit results and the last column are not significantly different, unrelated token regularization results in uneven attention scores among related tokens, while related token regularization leads to equal attention scores.

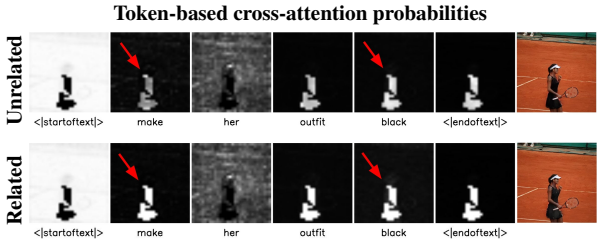

Figure 16. **Ablation study on related token rewarding.** Instead of regularizing the unrelated tokens, see the first row, we ablate the opposite which is rewarding the related tokens by giving + ∞ within RoI, see the second row.

A.7.2 Clustering Method Alternatives

Since our implementation uses KMeans, it requires a *number of clusters* parameter to be decided and fixed. While in the main paper, we show that a good tuning of this hyperparameter works robustly across all the evaluated datasets, in this section, we perform a preliminary exploration of alternative clustering techniques that would relax this assumption. We replace K-means with Agglomerative Clustering, which does not require this parameter. For this implementation, we use the cosine similarity metric between features since [46] shows that the cosine similarity provides significant information about semantics. We provide a minimum distance threshold, which can be between 0-1, and it automatically determines the number of clusters. The higher the threshold, the more clusters. As seen in Tab. 8, this variation achieves similar performance by properly tuning the distance threshold. This ablation study proves that our method is robust for selecting the clustering method, and any clustering method can be combined with LIME.

Table 8. **Ablation Study on Clustering Method.** For all experiments, IP2P is the base architecture and the evaluation is on the MagicBrush dataset. Each parameter is modified separately, while other parameters are kept fixed to isolate their impact.

	Method	L1 ↓	L2 ↓	CLIP-I ↑	DINO ↑	CLIP-T ↑
	IP2P [7]	0.112	0.037	0.852	0.743	0.276
# of Clusters	4	0.080	0.022	0.923	0.885	**0.295**
	8	**0.058**	**0.017**	**0.935**	**0.906**	0.293
	16	0.062	0.018	0.933	0.903	0.294
	32	0.064	0.018	0.932	0.901	0.291
Distance Threshold	0.7	0.080	0.022	0.927	0.893	**0.294**
	0.6	0.076	0.021	0.929	0.894	**0.294**
	0.5	**0.063**	**0.018**	**0.933**	**0.902**	0.293
	0.4	0.072	0.020	0.930	0.896	0.293

A.8. More Qualitative Results

This section presents additional qualitative results derived from our method, emphasizing its improved effectiveness against established baselines, such as IP2P [7] and IP2P w/MB [53]. Figure 17 illustrates the application of our method in localized image editing tasks. Specifically, it demonstrates our method's proficiency in altering the color of specific objects: (a) *ottoman*, (b) *lamp*, (c) *carpet*, and (d) *curtain*. Unlike the baseline methods, which tend to entangle the object of interest with surrounding elements, our approach achieves precise, disentangled edits. This is not achieved by the baseline, which tends to alter multiple objects simultaneously rather than isolating changes to the targeted region. The disentangled and localized edits showcased in Fig. 17 highlight the potential of LIME in end-user applications where object-specific edits are crucial.

Figure 19 demonstrates additional examples of our

Figure 17. **A use-case of the proposed method.** Changing the color of different objects is shown by comparing baselines and our method. Our method performs disentangled and localized edits for different colors and different objects in the scene.

method's performance on the Emu-Edit test set [44], the MagicBrush [53] test set and the PIE-Bench [21] dataset. Our approach effectively executes various tasks, such as (a) replacing an animal, (b) modifying parts of animals, and (c) changing the color of multiple objects. As illustrated in Fig. 19, our method demonstrates significant improvements over existing baselines. For instance, while baseline models like IP2P w/MB in (a) achieve reasonable edits, they often inadvertently modify areas outside the RoI, as observed in cases (b) and (c). Notably, our method helps focus the baseline models on the RoI, as seen in (b) and (c), where baselines struggle to preserve the original image. Although our method is dependent on the baseline and may occasionally induce unintended changes in peripheral areas, *e.g.*, the floor's color, it consistently outperforms the baseline models in terms of targeted and localized editing.

B. Implementation Details

We obtain the results using an NVIDIA A100 40GB GPU machine. For 512×512 images the IP2P-based baselines (*e.g.*, IP2P, IP2P w/MB, HIVE, and HIVE w/MB) take approximately 15 seconds per edit, while for LIME integrated models, it takes ≈25 seconds.

B.1. User Study Setting

We carry out a study with 54 questions involving users, asking 53 anonymous individuals on the crowd-sourcing platform Prolific [38]. In our user study, participants will be presented with two alternative edited images alongside their corresponding input images and editing instructions. They will be tasked with evaluating the effectiveness of the edits in achieving the specified outcome and the ability of the editing method to preserve the details in areas not targeted by the instruction. Using the example provided in Fig. 18, where the editing instruction is to *Change to a rosé*, participants must discern which edited image (a, b or neither) not only best satisfies this directive but also maintains the fidelity of the scene's irrelevant aspects. The aggregated data from participant responses will yield insights into the preferred methods for both accurate editing and detail preservation, thereby influencing the development of advanced image editing methods.

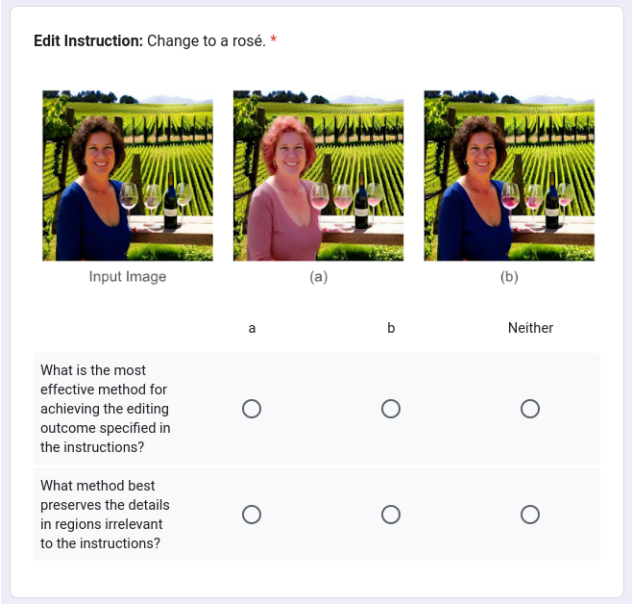

Figure 18. **User study setting.** The example with the edit instruction with the corresponding input image and randomly ordered two edited images from baseline and baseline + LIME.

B.2. Reproducibility Statement

We guarantee that all the results presented in the main manuscript and the supplementary materials can be reproduced. We will make our code base containing examples and hyperparameters public so that the results can be reproduced.

B.3. Baselines

Open-Edit [27]: This GAN-based approach uses a reconstruction loss for pre-training and incorporates a consistency loss during fine-tuning on specific images. Its unique feature is the arithmetic manipulation of word embeddings within a shared space of visual and textual features.
VQGAN-CLIP [10]: Enhancing the VQGAN [13] framework with CLIP embeddings [39], this method fine-tunes VQGAN using the similarity of CLIP embeddings between the generated image and the target text, leading to optimized image generation.
SDEdit [29]: Leveraging the capabilities of Stable Diffusion [41], SDEdit introduces a tuning-free approach. It uses stochastic differential equation noise, adding it to the source image and subsequently denoising to approximate the target image, all based on the target caption.
Text2LIVE [4]: It propose a Vision Transformer [12] for generating edited objects on an additional layer. It incorporates data augmentation and CLIP [39] supervision, ultimately alpha-blending the edited layer with the original to create the target image.
Null Text Inversion [33]: By optimizing the DDIM [45] trajectory, this method initially inverts the source image. After, it performs image editing during the denoising process guided by cross-attention [18] between text and image.
SINE [55]: Real images are edited using model-based guidance and patch-based fine-tuning process.
DreamBooth [42]: It fine-tunes a diffusion model by learning special text tokens and adjusting model parameters on a set of images for editing.
Textual-Inversion [15]: It fine-tunes a token embedding within the text-encoder space using a set of images.
Imagic [22]: It edits images through a three-step process: first fine-tuning a token embedding, then fine-tuning the parameters of a text-guided image diffusion model using the fine-tuned token embedding, and finally performing interpolation to generate various edits based on a target prompt.
DiffEdit [9]: It identifies the region to edit in images by contrasting between a conditional and unconditional diffusion model based on query and reference texts. Then, it reconstructs the edited image by collecting the features from the text-query by combining the features in the noise/latent space, considering the region to edit.
Blended Latent Diffusion [2]: This method uses a text-to-image Latent Diffusion Model (LDM) to edit the user-defined mask region. It extracts features for the mask region from the edit text, and for the rest of the image, it uses features from the original image in the noise/latent space.
DirectDiffusion [21]: It inverts the input image into the latent space of Stable Diffusion [41] and then applies Prompt2Prompt [18] to obtain the desired edit without making any changes to the edit diffusion branch.
Diffusion Disentanglement [50]: It finds the linear combination of the text embeddings of the input caption and the desired edit to be performed. Since it does not fine-tune Stable Diffusion parameters, they claim that the method performs disentangled edits.
InstructPix2Pix (IP2P) [7]: Starting from the foundation of Stable Diffusion [41], the model is fine-tuned for instruction-based editing tasks. It ensures that the edited image closely follows the given instructions while maintaining the source image without the need for test-time tuning.
InstructPix2Pix w/MagicBrush [53]: A version of IP2P [7] trained on MagicBrush train set [53]. Since the MagicBrush dataset has more localized edit examples, the fine-tuned version has better results, as seen in Fig. 6a.
HIVE [54]: It extends IP2P [7] by fine-tuning it with an expanded dataset. Further refinement is achieved through fine-tuning with a reward model, which is developed based on human-ranked data.
HIVE w/MagicBrush [53]: HIVE [54] fine-tuned on MagicBrush train set [53]. Since the MagicBrush dataset has more localized edit examples, the fine-tuned version has better results, as seen in Fig. 6a.

C. Broader Impact & Ethical Considerations

The advancement in localized image editing technology holds significant potential for enhancing creative expression and accessibility in digital media and virtual reality applications. However, it also raises critical ethical concerns, particularly regarding its misuse for creating deceptive imagery like deepfakes [24] and the potential impact on job markets in the image editing sector. Ethical considerations must focus on promoting responsible use, establishing clear guidelines to prevent abuse, and ensuring fairness and transparency, especially in sensitive areas like news media. Addressing these concerns is vital for maximizing the technology's positive impact while mitigating its risks.

Figure 19. **More Qualitative Examples.** We test our method on different tasks: (a) replacing an animal, (b) editing animal parts, and (c) changing the color of multiple objects. The integration of LIME enhances the performance of all models, enabling localized edits while maintaining the integrity of the remaining image areas.

References

[1] Aishwarya Agarwal, Srikrishna Karanam, K J Joseph, Apoorv Saxena, Koustava Goswami, and Balaji Vasan Srinivasan. A-star: Test-time attention segregation and retention for text-to-image synthesis. In *ICCV*, pages 2283–2293, October 2023. 2

[2] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM TOG*, 42(4):1–11, 2023. 4, 8

[3] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, pages 18187–18197. IEEE, 2022. 1, 2, 3, 4, 8

[4] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *ECCV*, volume 13675 of *Lecture Notes in Computer Science*, pages 707–723. Springer, 2022. 7, 8

[5] Samyadeep Basu, Mehrdad Saberi, Shweta Bhardwaj, Atoosa Malemir Chegini, Daniela Massiceti, Maziar Sanjabi, Shell Xu Hu, and Soheil Feizi. Editval: Benchmarking diffusion based text-guided image editing methods. *arXiv preprint arXiv:2310.02426*, 2023. 1, 3

[6] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009. 4

[7] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, pages 18392–18402, 2023. 1, 2, 3, 5, 7, 8, 6, 9

[8] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM TOG*, 42(4):1–10, 2023. 2, 4

[9] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. In *ICLR*, 2023. 1, 2, 3, 4, 8

[10] 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 *ECCV*, volume 13697 of *Lecture Notes in Computer Science*, pages 88–105. Springer, 2022. 7, 4, 8

[11] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 34:8780–8794, 2021. 3

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 8

[13] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pages 12873–12883, 2021. 8

[14] 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 *ICLR*, 2024. 2

[15] 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 *ICLR*, 2023. 1, 3, 8

[16] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Houqiang Li, Han Hu, et al. Instructdiffusion: A generalist modeling interface for vision tasks. In *CVPR*, pages 12709–12720, 2024. 1, 2, 3

[17] Qin Guo and Tianwei Lin. Focus on your instruction: Fine-grained and multi-instruction image editing by attention modulation. *arXiv preprint arXiv:2312.10113*, 2023. 2, 3

[18] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-prompt image editing with cross-attention control. In *ICLR*, 2023. 1, 2, 5, 8

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 1

[20] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 3

[21] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. In *ICLR*, 2024. 2, 1, 3, 4, 7, 8

[22] 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 *CVPR*, pages 6007–6017, 2023. 1, 3, 8

[23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. *CoRR*, abs/2304.02643, 2023. 5

[24] Pavel Korshunov and Sébastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018. 8

[25] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *CVPR*, pages 7061–7070, 2023. 5

[26] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014. 3

[27] 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 *ECCV*, volume 12356 of *Lecture Notes in Computer Science*, pages 89–106. Springer, 2020. 7, 8

[28] Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, pages 11451–11461. IEEE, 2022. 1

[29] 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 *ICLR*, 2022. 1, 2, 7, 3, 8

[30] Tuna Han Salih Meral, Enis Simsar, Federico Tombari, and Pinar Yanardag. Conform: Contrast is all you need for high-fidelity text-to-image diffusion models. In *CVPR*, 2024. 2

[31] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection. In *Computer Vision – ECCV 2022: 17th European Conference, 2022*, page 728–755, 2022. 3

[32] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Marcus A Brubaker, Jonathan Kelly, Alex Levinshtein, Konstantinos G Derpanis, and Igor Gilitschenski. Watch your steps: Local image and scene editing by text instructions. *arXiv preprint arXiv:2308.08947*, 2023. 3, 4, 8

[33] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, pages 6038–6047, 2023. 2, 7, 3, 8

[34] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, volume 162, pages 16784–16804, 2022. 1

[35] 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*, pages 1–11, 2023. 2

[36] Or Patashnik, Daniel Garibi, Idan Azuri, Hadar Averbuch-Elor, and Daniel Cohen-Or. Localizing object-level shape variations with text-to-image diffusion models. In *ICCV*, 2023. 1, 3, 8, 5

[37] Koutilya PNVR, Bharat Singh, Pallabi Ghosh, Behjat Siddiquie, and David Jacobs. Ld-znet: A latent diffusion approach for text-based image segmentation. In *ICCV*, pages 4157–4168, October 2023. 2, 3, 4, 7

[38] Prolific. https://www.prolific.com/. Accessed: 2024-01-24. 8, 7

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, volume 139, pages 8748–8763, 2021. 4, 7, 8

[40] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022. 1

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10674–10685, 2022. 1, 3, 5, 8

[42] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, pages 22500–22510, June 2023. 1, 3, 8

[43] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022. 1

[44] 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. *arXiv preprint arXiv:2311.10089*, 2023. 7

[45] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th ICLR, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 8

[46] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. In *NeurIPS*, volume 36, pages 1363–1389, 2023. 2, 3, 6

[47] 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 *NeurIPS*, 2023. 2

[48] Qian Wang, Biao Zhang, Michael Birsak, and Peter Wonka. Mdp: A generalized framework for text-guided image editing by manipulating the diffusion path, 2023. 2

[49] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation. In *ICCV*, pages 15943–15953, 2023. 1

[50] Qiucheng Wu, Yujian Liu, Handong Zhao, Ajinkya Kale, Trung Bui, Tong Yu, Zhe Lin, Yang Zhang, and Shiyu Chang. Uncovering the disentanglement capability in text-to-image diffusion models. In *CVPR*, pages 1900–1910, 2023. 2, 4, 8

[51] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, pages 2955–2966, 2023. 2, 3, 5

[52] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *CVPR*, pages 18381–18391, 2023. 1

[53] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In *NeurIPS*, 2023. 1, 2, 5, 7, 8, 3, 6, 9

[54] 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 *CVPR*, pages 9026–9036, 2024. 1, 2, 5, 3, 8

[55] Zhixing Zhang, Ligong Han, Arnab Ghosh, Dimitris Metaxas, and Jian Ren. Sine: Single image editing with text-to-image diffusion models, 2022. 3, 8

[56] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *NeurIPS*, 36, 2024. 5