AnyEdit: Mastering Unified High-Quality Image Editing for Any Idea

Qifan Yu[1*] Wei Chow[1*] Zhongqi Yue[2] Kaihang Pan[1] Yang Wu[3] Xiaoyang Wan[1]
Juncheng Li[1†] Siliang Tang[1] Hanwang Zhang[2] Yueting Zhuang[1†]
[1]Zhejiang University, [2]Nanyang Technological University, [3]Ant Group
{yuqifan, 3210103790, kaihangpan, junchengli, siliang, yzhuang}@zju.edu.cn
nickyuezhongqi@gmail.com, wy306396@antgroup.com

https://dcd-anyedit.github.io/

Abstract

Instruction-based image editing aims to modify specific image elements with natural language instructions. However, current models in this domain often struggle to execute complex user instructions accurately, as they are trained on low-quality data with limited editing types. We present ***AnyEdit****, a comprehensive multi-modal instruction editing dataset, comprising 2.5 million high-quality editing pairs spanning over 20 editing types and five domains. We ensure the diversity and quality of the AnyEdit collection through three aspects: initial data diversity, adaptive editing process, and automated selection of editing results. Using the dataset, we further train a novel AnyEdit Stable Diffusion with task-aware routing and learnable task embedding for unified image editing. Comprehensive experiments on three benchmark datasets show that AnyEdit consistently boosts the performance of diffusion-based editing models. This presents prospects for developing instruction-driven image editing models that support human creativity.*

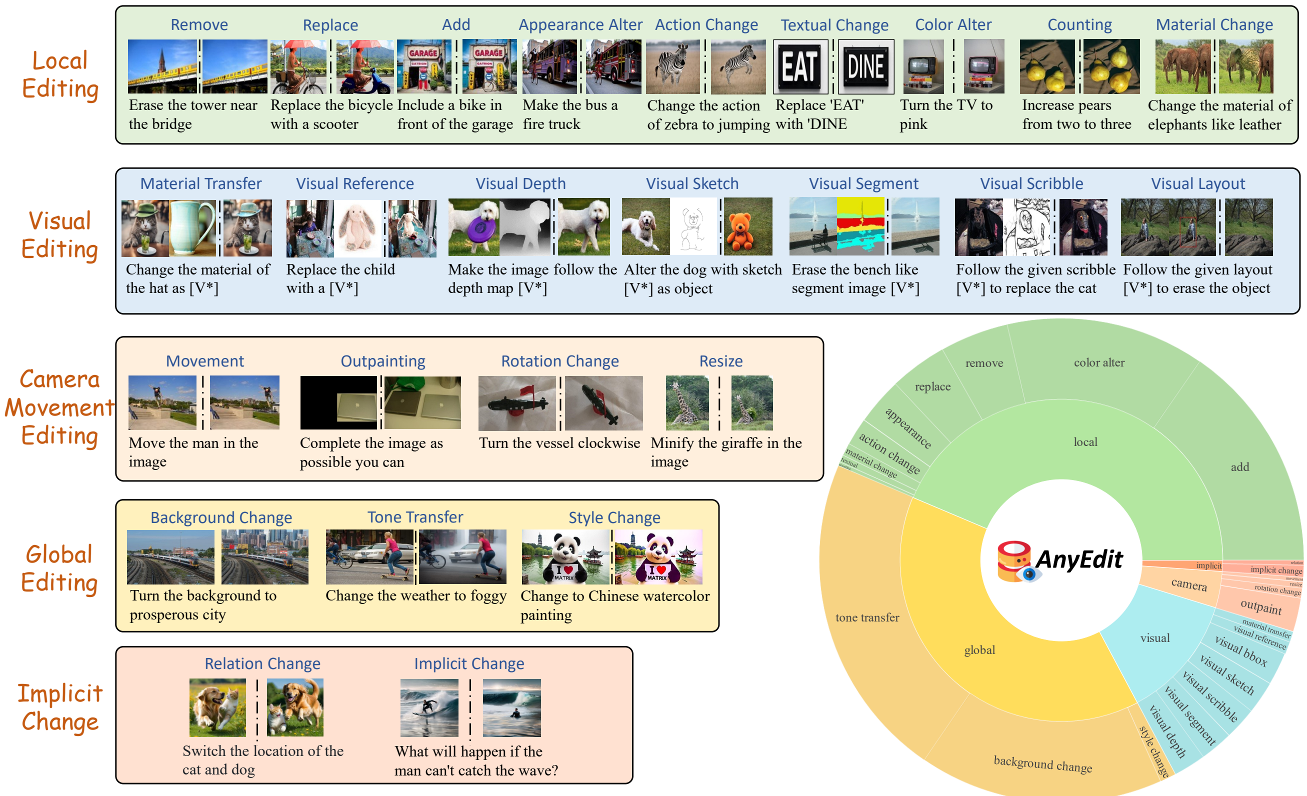
1. Introduction

With the development of multi-modal datasets [26, 40] and generative frameworks [11, 17], recent text-to-image generation (T2I) models [2, 3, 32, 37–39, 48, 51] have shown promising results in generating high-fidelity photo-realistic images. To enhance the controllability of generated images, image editing [7, 10, 19, 27, 42, 61] aims to modify desired target elements and retain unrelated contents, which has achieved significant improvements for T2I models. More recently, instruction-based image editing [5, 6, 10, 19, 48] pave the way to conveniently edit images using natural language instructions without complex descriptions or region-specific masks. However, the scarcity of high-quality instruction editing datasets makes it difficult to develop powerful instruction-guided image editing models.

Despite several prior works [5, 14, 15, 41, 49, 54, 57] showcasing promising instruction-following editing datasets and achieving effective results, they still struggle with drawbacks due to limited editing types and low data quality. These methods lack support for editing instructions based on complex multi-modal perception (*e.g.*, spatial composition [59], viewpoint [53], commonsense understanding [4]) and reference-based editing instructions (*e.g.*, customization [8, 47], image transfer [55]), limiting the model's ability to understand interleaved editing instructions and robustly execute a wide variety of editing tasks. To address the above limitations, this work, **AnyEdit**, first provides a comprehensive catalog of editing instructions and systematically introduces a unified editing framework for various editing tasks. As shown in figure 1, AnyEdit is divided into five classes based on editing capabilities, with each class containing multiple task types. Regarding the general editing scenarios in previous work [15, 54, 57], AnyEdit categorizes them into **Local Editing** and **Global Editing** based on their granularity to construct more precise instructions. To accommodate the editing intentions for spatial perception, we further include **Camera Movement Editing**. For commonsense understanding, we introduce **Implicit Editing**, which replaces direct intentions with more complex instructions that require imaginative thinking. In addition, to accommodate reference inputs from the visual modality, we incorporate **Visual Editing** to construct multi-modal editing instructions.

Although AnyEdit has enriched editing types, there exist three principal challenges for high-quality dataset collection: (1) First, we observe that current editing datasets overlook inherent data biases when preparing initial image-text pairs [13, 31, 50], leading to an imbalance of concepts within the dataset. For instance, "*change the tree in the*

*Equal Contribution.
†Corresponding Author.



(a) AnyEdit Dataset Collection Visualization Examples

(b) AnyEdit Dataset Collection Distribution

Figure 1. Examples of **AnyEdit** at scale. We comprehensively categorize image editing tasks into 5 groups based on different editing capabilities: (a) *Local Editing* which focuses on region-based editing (green area); (b) *Global Editing* which focuses on the full range of image rendering (yellow area); (c) *Camera Move Editing* which focuses on viewpoints changing instead of scenes (gray area); (d) *Implicit Editing* which requires commonsense knowledge to complete complex editing (orange area); (e) *Visual Editing* which encompasses additional visual inputs, addressing the requirements for multi-modal editing (blue area).

park to a camel" may bring undesired sand because "sand" and "camel" frequently co-occur in the training data. Thus, we introduce counterfactual synthetic scenes to enrich real-world data with tail concepts and their diverse combinations, achieving a balanced distribution of concepts in the dataset and enhancing generalization in editing tasks. (2) Secondly, current data collection methods struggle to accommodate diverse input formats and editing requirements due to the shared pipeline for all instructions. Therefore, we propose an adaptive editing pipeline that automatically selects the appropriate data pipeline for each task. (3) Third, there exist inherent drawbacks in the collected triplets of original images, edit instructions, and edited images due to the randomness of generation (*i.e.*, low resolution [40], high noise [18], and misalignment between text and images [43, 45]). Thus, we develop an instruction validation pre-filter and image assessment post-filter strategy to filter out unsatisfactory results. In this way, AnyEdit automatically comprises 2.5M high-quality editing data across 25 distinct editing types (c.f., Tab. 1), showing a 28.9% improvement in visual similarity and 18.8% in semantic similarity over SOTA datasets (c.f., Tab. 3). This automated process opens up the possibility of scaling the high-quality editing dataset in a low-resource manner.

To fully harness the potential of AnyEdit's high-quality editing data, we propose a novel AnyEdit Stable Diffusion approach (**AnySD**) that employs task-aware routing and learnable task embeddings to support various editing types within AnyEdit, thereby constructing a robust instruction-based editing model for handling diverse editing requests. The task-aware routing mechanism leverages AnyEdit's diverse data to adjust the granularity of editing (*e.g.*, local object or global style), while the learnable task embedding coordinates the inherent complexity across tasks within AnyEdit. In this way, AnySD fully harnesses the comprehensive and diverse data from AnyEdit to set new records on the MagicBrush and Emu-Edit benchmarks.

Furthermore, to facilitate a comprehensive evaluation of instruction-based image editing, we manually curate a challenging benchmark, **AnyEdit-Test**, from our automatically collected AnyEdit dataset, comprising 1,250 high-quality editing pairs in diverse scenarios, across 25 distinct cate-

Editing Dataset	#Size	#Types	Instruction Type					Target Scenario			Open Source
			Local	Global	Camera Movement	Implicit	Visual	Real Image	Synthetic Image	Synthetic Scene	
MagicBrush [54]	10K	5	✓	✓	×	×	×	✓	×	×	✓
InstructPix2Pix [5]	313K	4	✓	✓	×	×	×	×	✓	×	✓
EMU-Edit [41]	-	8	✓	✓	×	✓	×	×	✓	×	×
HQ-Edit [21]	197K	6	✓	✓	×	×	×	×	✓	×	✓
SEED-Data-Edit [15]	3.7M	6	✓	✓	✓	×	×	✓	✓	×	✓
EditWorld [49]	8.6K	1	×	×	×	✓	×	✓	×	×	✓
UltraEdit [57]	4M	9	✓	✓	×	×	×	✓	✓	×	✓
AnyEdit	2.5M	**25**	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1. Comparison of existing image editing datasets. "Real Image" means the original images are from real world, "Synthetic Image" means they are from T2I models, "Synthetic Scene" indicates both images and captions are generated to address the inherent data bias.

gories (50 per category). We notice that (1) Existing models often fail to properly modify or cause distortions in complex tasks (*e.g.*, action change shown in Fig. 5), exposing the limitations of current benchmarks for complex tasks. (2) Even for common tasks in AnyEdit-Test, some previous SOTA models show a notable performance drop compared to existing benchmarks (*e.g.*, DINO's decline in global editing in Tab. 5), revealing the limitations of current benchmarks in multi-scene editing. This highlights the critical role of AnyEdit-Test in thoroughly assessing the capabilities of editing models. We will release it for public research.

Our main contributions are summarized as follows:

- We systematically categorize various editing instructions in a novel perspective and innovatively introduce a unified editing framework that leverages an adaptive editing pipeline to automatically collect diverse high-quality editing data across different scenarios in a scalable way.
- We construct a multi-type, multi-scene dataset, AnyEdit, and its corresponding benchmark AnyEdit-Test for instruction-based editing in various scenarios, comprising 25 distinct and complex editing types to accommodate broader requirements of real-world editing.
- We use our proposed AnySD to fully unlock the potential of AnyEdit, achieving SOTA improvements in instruction adherence and image fidelity across diverse editing types.

2. Related Work

Table 1 compares the existing Instruction-based image editing datasets including InstructPix2Pix [5], MagicBrush [54], EMU-Edit [41], HQ-Edit [21], SEED-Data-Edit [15], EDITWORLD [49], and UltraEdit [57]. Among these, the first four datasets use automatic methods to collect data, while the latter three incorporate human curation. InstructPix2Pix [5] utilizes P2P [19] for image editing, which makes it only suitable for synthetic images. MagicBrush [54] hires crowd workers to annotate images from the MSCOCO [26] dataset manually but only includes 10K editing pairs due to expensive labor expenses. HQ-Edit[21] also lacks fine-grained details and realism due to its diptych generation though it exploits GPT-4V [1] and DALL-E [3] to enhance descriptions. Both of them are lack of generalization. With the advancements in multi-modal understanding [22–25, 33–35, 52] and generation [9, 16, 36, 46],

Dataset	#Samples	#Concepts	License	Annotator
Real World Image Caption Paired Dataset				
MS COCO [26]	123,287	80	CC BY 4.0	Human
MVImgNet [53]	31,783	238	CC BY 4.0	Human
LLaVA-CC3M-Pretrain [28]	519,176	31423	Custom	GPT-4V
Counterfactual Synthetic Scene Pair Dataset				
AnyEdit-Composition	22,936	500	Custom	Composition

Table 2. Data preparation details for AnyEdit dataset collection.

SEED-Data-Edit [15] and UltraEdit [57] have been recently introduced to enrich the types and turns of image editing. However, they only focus on traditional editing tasks, neglecting more complex and customized editing needs. Although EMU-Edit [41] and EditWorld [49] endeavor to simulate physical world editing, they lack a unified definition of implicit editing. Moreover, these datasets fail to incorporate viewpoint changes and support multi-modal inputs. In AnyEdit, we combine five distinct groups of data, covering 25 editing types, which will be released to help the community. It is worth noting that AnyEdit is the only dataset that considers the data bias and introduces counterfactual synthetic scenes to balance the distribution of the dataset.

3. AnyEdit

3.1. Editing Type Definition

To equip instruction-based editing models with comprehensive capabilities to follow any creative ideas, we compiled a multi-modal image editing dataset **AnyEdit** for instruction-based image editing, consisting of 2.5M high-quality editing pairs across five primary domains, as illustrated in Fig. 1. The dataset comprises editing tasks divided into five main categories, each containing various editing types: (1) Local editing: add, remove, replace, color alter, appearance alter, material change, action change, textual change, and counting. (2) Global editing: background change, tone transfer, and style change. (3) Camera movement editing: rotation change, out-painting, movement, and resize. (4) Implicit editing: implicit change and relation change. (5) Visual editing: visual reference, material transfer, and visual conditions (*i.e.*, depth, segmentation, scribble, sketch, mask). Specifically, local editing targets specific areas of an image without altering unrelated semantic content, while global editing affects the entire image. Camera movement editing extends this concept by manipulating the viewpoint of specific objects or the whole content within the scene. Furthermore, implicit editing involves hidden changes in

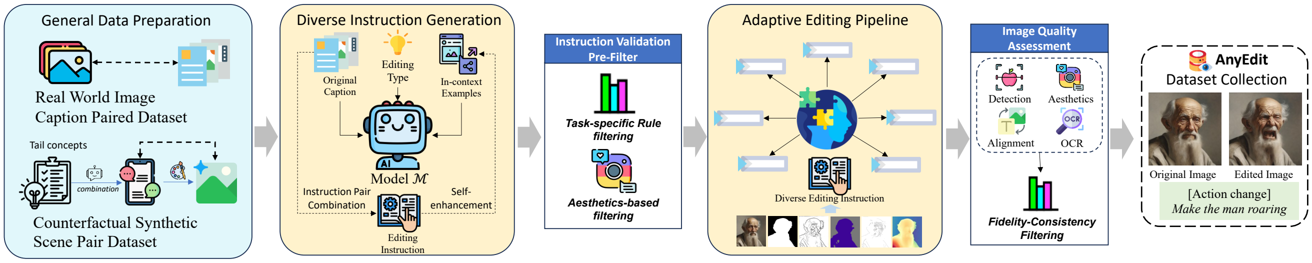


Figure 2. The comprehensive construction pipeline of AnyEdit. We summarize the general pipeline into five steps: (1) General data preparation from real-world image-text pairs and synthetic scenes. (2) Diverse instruction generation using LLM to produce high-quality editing instructions. (3) Pre-filtering for instruction validation. (4) Adaptive editing pipeline tailors specific editing methods for each edit type to generate high-quality edited images. (5) Image quality assessment ensures high-quality editing pairs for the AnyEdit Dataset.

state or interaction patterns that require comprehension. Visual editing, on the other hand, incorporates additional visual inputs as references alongside editing instructions. Figure 1(a) shows examples of various editing types in each category and their detailed definitions are in Appendix B.1.

3.2. Automatic Dataset Collection

General Data Preparation. Previous studies have demonstrated that high-quality initial images facilitate the diversity of editing image creation [15, 57]. To address real-world user demands for image editing in complex scenarios, we collect ~680K real-world image-caption pairs from annotated datasets (*i.e.*, MSCOCO [26], LLaVA-CC3M-Pretrain [28]) and multi-view image datasets (*i.e.*, MVImgNet [53]). Then, we enrich those brief captions by MLLMs (*e.g.*, VILA [25]) for the completeness of the descriptions. However, these image-text pairs suffer from inherent data bias, leading to the model falling short in domains not well covered by general knowledge [13, 31]. Thus, we introduce the Counterfactual Synthetic Scene Pair Dataset to balance the data distribution of the initial image-text pairs. Specifically, we collect infrequent tail concepts from internet data and combine multiple concepts and contexts to generate a description by LLaMA-3B [12]. Subsequently, we invoke off-the-shelf T2I models to produce the initial images. In this manner, we enrich the original dataset by incorporating rare concept combinations, resulting in ~700K high-quality and diverse image-caption pairs for the AnyEdit dataset collection, as illustrated in Table 2.

Diverse Instruction Generation. It aims to create varied editing instructions and corresponding edited caption outputs based on the caption of the initial image. As illustrated in Figure 2, we leverage the public Llama3-8b [12] model to convert original captions to diverse editing instructions. To address the limitations in instruction diversity and consistency when generating editing instructions, we integrate intuitive type constraints with LLM generation and employ in-context examples to develop a task-specific agent tailored to each editing type. Furthermore, we integrate the generated editing instructions with the original captions to create instruction pairs, which serve as in-context examples for iterative self-enhancement, thereby gradually increasing the diversity and complexity of the instructions. Details of the prompt constraints and examples are in Appendix B.2.

Adaptive Editing Pipelines. Traditional instruction-editing datasets [41, 57] rely on fixed pipelines [19] or time-consuming manual filtering, making it difficult to efficiently generate high-quality edited images for complex editing types and various input formats. Here we propose an adaptive view of editing pipelines that customizes the edited image according to the specific type of editing. Specifically, we design 9 core pipelines for generating local, global, camera movement, implicit, and visual editing data. During image editing generation, we input editing instruction pairs along with the original images and their variants into the adaptive editing pipeline. The pipeline dynamically selects tailored solutions based on the editing type to generate images that align with the intended edits. Additionally, we incorporate extra constraints (*e.g.*, dilated masks, layouts, and geometry guidance) into the U-Net layers within the diffusion process to achieve more precise semantic alignment and artifact reduction. Full details of the image construction process for each editing type can be found in Appendix B.3.

Data Quality Enhancement. Since the quality of editing data is critical for robust editing model training in AnyEdit, we further introduce a comprehensive filtering strategy for data quality enhancement. It consists of two steps: *instruction validation pre-filter* and *image quality post-filter*.

(i) Instruction Validation Pre-filter. We notice that partial editing instructions from LLMs sometimes introduce ambiguities that adversely impact the edited image (*e.g.*, changing appearance in "color alter" editing or changing action of the static desk in "action change" editing). Simultaneously, low-quality initial images (*e.g.*, low resolution, bad aspect ratio, lack of aesthetics) consistently lead to unsatisfactory editing results, even after time-consuming rounds of selection. Thus, we employ task-specific heuristic rules to validate various instructions, ensuring consistency and performing aesthetic evaluations to guarantee that aesthetically

Dataset	Semantic Similarity		Visual Similarity	
	CLIPin	CLIPout	SSIM	DINOv2
IP2P [5]	0.2650	0.2694	0.5826	0.6859
UltraEdit [57]	0.2834	0.3049	0.6401	0.7231
AnyEdit (Ours)	**0.3288**	**0.3133**	**0.6638**	**0.9053**

Table 3. Overall dataset quality comparison between AnyEdit and other datasets using automatic semantic and visual metrics.

balanced images are used for the editing process.

(ii) Image Quality Post-filter. After collecting all the editing results, we conducted an automatic data evaluation to filter out unsatisfactory images that did not meet the criteria for data generation. First, we utilize CLIP filtering metrics [41, 57] to assess the alignment between edited images I_e and their output captions T_e, ensuring consistency with editing instructions and desired modifications in target regions. Subsequently, we utilize CLIP image similarity to compare the original image I_o and the edited image I_e, ensuring the fidelity of unedited image elements at the pixel level. Additionally, we compute the L1 distance between I_o and I_e to maintain the global structure of the original images. Finally, we employ CLIP directional similarity [54], which quantifies the alignment between modifications in images and the associated changes in captions, to verify the AnyEdit's instruction-following capability. We further apply the object detector to validate the presence of edited objects (in Local Editing) and the VLM to assess the modification of global images (in Global Editing).

3.3. Characteristics and Statistics

Benefiting from our promising automated dataset collection approach, AnyEdit comprises 2.5 million high-quality editing pairs across 25 distinct editing types. AnyEdit encompasses a broader range of domains, including complex editing tasks such as viewpoint editing, implicit editing, and visual editing, and it incorporates a richer variety of scenes, including conceptually rich synthetic scenes (c.f., Tab. 1). Moreover, the data distribution of AnyEdit in Fig. 1(b) reflects a broad variety of edit types, with coverage across domains. Quantitatively, we assess AnyEdit's data quality based on semantic similarity and visual similarity metrics (c.f., Tab. 3). The significant improvement (+25.2% in DINOv2 and +16.0% in CLIPin compared to the UltraEdit) shows AnyEdit's effectiveness in maintaining pixel-level consistency and accurately reflecting editing instructions.

4. Method

4.1. Architecture

Since AnyEdit contains a wide range of editing instructions across various domains, it holds promising potential for developing a powerful editing model to address high-quality editing tasks. However, training such a model has

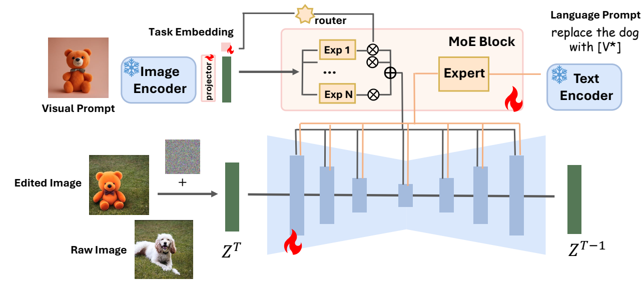


Figure 3. **Architecture of AnySD**. AnySD is a novel architecture to supports three conditions (original image, editing instruction, visual prompt) for various editing tasks.

three extra challenges: (a) aligning the semantics of various multi-modal inputs; (b) identifying the semantic edits within each domain to control the granularity and scope of the edits; (c) coordinating the complexity of various editing tasks to prevent catastrophic forgetting. To this end, we propose a novel AnyEdit Stable Diffusion approach (**AnySD**) to cope with various editing tasks in the real world. As illustrated in Figure 3, AnySD includes three designs: visual prompt projector, task-aware routing, and learnable task embedding. Then, we will introduce each AnySD design.

Visual Prompt Projector. To align the semantics of multi-modal inputs, we first use a visual prompt projector to align these image features z_V from the frozen CLIP image encoder with the instruction embeddings to obtain visual prompt c_V. We then integrate these embeddings into the timestep embeddings of the UNet through cross-attention interactions to provide visual prompt conditioning during general image editing diffusion [5]. The diffusion objective is then adjusted as follows:

$$L = \mathbb{E}_{\mathcal{E}(x),\mathcal{E}(c_I),c_T,c_V,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon-\epsilon_\theta(z_t,t,\mathcal{E}(c_I),c_T,c_V))\|_2^2\right] \tag{1}$$

where $\epsilon \in N(0,1)$ is the noise added by the diffusion process and $[x, c_I, c_T, (c_V)]$ is a quadruple of editing instruction, with c_V only being applicable for visual editing.

Task-aware Routing. Since each editing type exhibits different granularities and scopes of edits, we propose a task-aware routing strategy with Mixture of Experts (MoE) [29] blocks to meet task-specific editing requirements. Specifically, the visual condition c_V is integrated into the frozen UNet layers with decoupled cross-attention to avoid disrupting the edit instruction condition. Each MoE block shares the same text attention layer but has diverse visual attention weights for c_V by the router based on the task embedding. Therefore, we introduce a cross-attention layer for each expert and add the distributed results from all experts to the output of text cross-attention as follows,

$$\mathbf{Z}^{new} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V} + \text{Softmax}\left(\frac{\mathbf{Q}(\mathbf{K}')^\top}{\sqrt{d}}\right)\mathbf{V}' \tag{2}$$

where $\mathbf{K}' = \boldsymbol{c}_v\mathbf{W}_k'$ is the query matrix from the visual

	EMU-Edit Test				MagicBrush Test			
	CLIPim ↑	CLIPout ↑	L1 ↓	DINO ↑	CLIPim ↑	CLIPout ↑	L1 ↓	DINO ↑
PnP [44]	0.521	0.089	0.089	0.304	0.568	0.101	0.289	0.220
Null-Text [30]	0.761	0.236	0.075	0.678	0.752	0.263	0.077	0.664
InstructPix2Pix [5]	0.834	0.219	0.121	0.762	0.837	0.245	0.093	0.767
EMU-Edit [41]	0.859	0.231	0.094	0.819	0.897	0.261	0.052	0.879
UltraEdit (SD3) [57]	0.845	0.283	0.071	0.794	-	-	-	-
SD1.5 w/ MagicBrush [54]	0.838	0.222	0.100	0.776	0.883	0.261	0.058	0.871
SD1.5 w/ AnyEdit (Ours)	0.866 (+3.3%)	0.284 (+27.9%)	0.095 (-5.0%)	0.812 (+4.6%)	0.892 (+1.0%)	0.273 (+4.6%)	0.057 (-1.7%)	0.877 (+0.7%)
AnySD w/ AnyEdit (Ours)	**0.872** (+4.1%)	**0.285** (+28.4%)	**0.070** (-30.0%)	**0.821** (+5.8%)	**0.898** (+1.7%)	**0.275** (+5.4%)	**0.051** (-12.1%)	**0.881** (+1.1%)

Table 4. Comparison of methods on EMU-Edit [41] and MagicBrush [54] benchmark. We show performance improvements over SOTA models of the same architecture, with only training data differences.

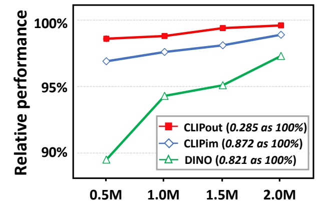


Figure 4. Dataset scales effect.

prompt and $\mathbf{W}'_k$ is the corresponding weight matrix.

Learnable Task Embeddings. To guide the generation process toward the appropriate granularity for each editing task, we learn a task embedding v_i during training. Unlike previous work [41], we insert task embeddings before the MoE block rather than integrate them into the whole denoising process to alleviate query confusion between instructions and editing types. The task embedding v_i has two main roles: (1) It concatenates with z_v (set to zero when no visual prompt is provided) to form c_V for MoE blocks; (2) it inputs to the router [60], distributing weights across MoE experts. More details of architecture are in Appendix G.1

4.2. Training and Inference

To enhance AnySD's ability to handle diverse conditioning for editing, we introduce CFG [20] extending from InstructPix2Pix [5] for three conditioning. Furthermore, we structure our training into two stages for AnySD, ensuring that the diffusion model can thoroughly understand general editing knowledge and develop fine-grained task-specific skills.

Stage I: Instruction Understanding. In this phase, we freeze the task-aware router and only pre-train the UNet layer to align with editing instructions. Additionally, We set additional conditions to zero tensors by CFG to enhance the model's instruction-following capability.

Stage II: Task Tuning. In this phase, we further fine-tune $\mathbf{W}'_k$ and $\mathbf{W}'_v$ within the MoE block, task-aware router, visual prompt projector, and task embeddings to adapt the model to the task-specific editing granularity.

Inference. In the inference stage, we predict the editing type with LLMs (*e.g.*, LLaMA-3) when given the input instruction. Then, we apply our AnySD for editing.

5. Experiments

In this section, we first assess AnyEdit and AnySD on popular standard editing benchmarks (§ 5.2), demonstrating the high quality of the AnyEdit dataset and the superiority of the AnySD architecture. Additionally, we extend the evaluation to the more challenging AnyEdit-Test benchmark (§ 5.3) to show the promising expandability of our approach, better aligning with the creative editing demands in real-world scenarios. We further present qualitative results (§ 5.4) and conduct in-depth analysis (§ 5.5) to illustrate the scalability and broader applicability of AnyEdit.

		CLIPim ↑	CLIPout ↑	L1 ↓	DINO ↑
Local	Null-Text [30]	0.773	0.270	0.245	0.641
	InstructPix2Pix [5]	0.753	0.274	0.164	0.562
	MagicBrush [54]	0.823	0.293	0.120	0.698
	HIVE^w [56]	0.814	0.294	0.184	0.651
	HIVE^c [56]	0.845	0.299	0.114	0.766
	UltraEdit (SD3) [57]	0.831	**0.308**	0.112	0.731
	AnyEdit (Ours)	**0.863**	0.297	**0.094**	**0.788**
Global	Null-Text [30]	0.753	0.277	0.270	0.613
	InstructPix2Pix [5]	0.747	0.261	0.180	0.523
	MagicBrush [54]	0.731	0.278	0.233	0.493
	HIVE^w [56]	0.762	0.287	0.196	0.579
	HIVE^c [56]	0.787	0.294	0.253	0.576
	UltraEdit (SD3) [57]	0.772	0.297	0.191	0.619
	AnyEdit (Ours)	**0.788**	**0.301**	**0.159**	**0.647**
Camera Movement	InstructPix2Pix [5]	0.700	-	0.178	0.477
	MagicBrush [54]	0.765	-	0.170	0.589
	HIVE^w [56]	0.779	-	0.175	0.619
	HIVE^c [56]	**0.838**	-	0.171	0.739
	UltraEdit (SD3) [57]	0.802	-	0.340	0.514
	AnyEdit (Ours)	0.833	-	**0.110**	**0.745**
Implicit	InstructPix2Pix [5]	0.794	0.288	0.190	0.558
	MagicBrush [54]	0.865	0.280	0.149	0.711
	HIVE^w [56]	0.821	0.284	0.161	0.635
	HIVE^c [56]	0.862	0.284	0.137	0.728
	UltraEdit (SD3) [57]	0.856	0.281	0.135	0.703
	AnyEdit (Ours)	**0.867**	**0.289**	**0.130**	**0.733**
Visual	Uni-controlnet [58]	0.717	0.249	0.260	0.442
	AnyEdit (Ours)	**0.801**	**0.258**	**0.145**	**0.628**

Table 5. Comparison of methods on AnyEdit-Test benchmark.

5.1. Experimental Setup

Settings. For a fair comparison, we adopt Stable-Diffusion 1.5 [38] as the backbone and follow the settings of InstructPix2Pix [5] to train our AnySD. Notably, we only use data from AnyEdit for training, without incorporating any additional datasets. More details are in Appendix G.5.

Benchmarks & Metrics. We access our method across two popular benchmarks: Emu Edit Test [41] and MagicBrush [54]. These standard benchmarks evaluate editing models by comparing edited results with ground truths. Additionally, we manually selected 50 high-quality editing data from each editing type in AnyEdit, creating AnyEdit-Test for a more challenging and comprehensive evaluation. Notably, AnyEdit-Test is not visible during training. See Appendix E for details of AnyEdit-Test. Following prior work [14, 41, 54], we adopt semantic similarity (*i.e.*,

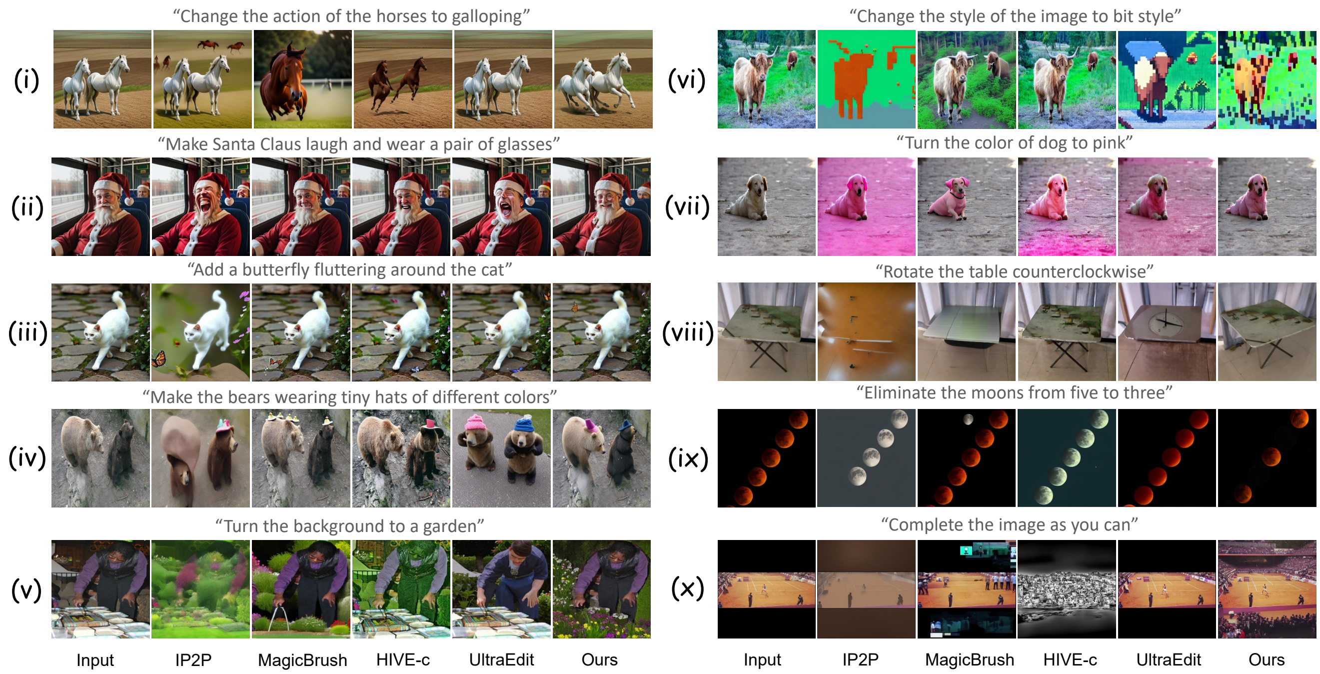


Figure 5. Qualitative evaluation of 10 distinct and complex tasks on AnyEdit-Test, such as *action change* (i), *appearance alter* (iv), *rotation change* (viii), *counting* (ix), and *outpainting* (x), demonstrates that our method yields promising results across these editing tasks.

CLIPim and CLIPout) and visual similarity (*i.e.*, DINO and L1 distance) metrics to evaluate the effectiveness of AnySD trained on AnyEdit for instruction-based image editing.

Baselines. We use the following baselines: 1) specialized image editing methods: PnP [44], Null-Text [30]. 2) instruction-based methods: it directly edits images with natural language, including InstructPix2Pix [5], MagicBrush [54], HIVE [56], EMU-Edit [41], UltraEdit [57]. 3) visual condition methods: it targets visual editing, including Uni-controlnet [58]. More details are in Appendix G.6.

5.2. Main Results on Standard Image Editing

We report the standard image editing results of AnyEdit and other baselines on EMU-Edit Test and MagicBrush benchmarks in Table 4. Based on the experimental results, we have summarized the following conclusions: ***(i)*** Our SD-1.5 with AnyEdit, which only changes the training data to AnyEdit, consistently demonstrates superior semantic performance in both edit alignment and content preservation compared to SOTA methods, even without additional mask supervision (0.872 for CLIPim and 0.285 for CLIPout on the EMU-Edit Test). It highlights AnyEdit's effectiveness in mastering high-quality image editing, validating its **high-quality editing data with significant semantic alignment and underlying clear editing structure**. ***(ii)*** Our AnySD model, trained on AnyEdit using the AnySD architecture, further surpasses SOTA methods in both semantic and visual similarity (0.872 of CLIPim on EMU-Edit Test and 0.881 of DINO on MagicBrush Test), setting new records on MagicBrush and Emu-Edit benchmarks. This demonstrates **the superiority of AnySD in following editing instructions while preserving unchanged image elements**, thanks to its task-aware architecture that learns task-specific knowledge from the diverse editing types in AnyEdit, enhancing the model's cross-task editing capabilities.

5.3. Comparison on AnyEdit-Test Benchmark

Table 5 presents the results of the AnyEdit-Test benchmark, where each instruction is designed to rigorously evaluate AnyEdit's adaptability across a wider range of challenging editing scenarios. We provide further results of each editing category in Appendix F. It can be observed that ***(i)*** most baselines struggle to effectively handle more complex editing tasks that are rarely in standard benchmarks (0.190 v.s. 0.121 on average L1), especially for implicit editing that requires reasoning abilities. This illustrates **the importance of AnyEdit-Test for evaluating the performance of editing models on complex tasks**. ***(ii)*** Even for common editing tasks, state-of-the-art models show a significant decline in consistency performance on AnyEdit-Test (-3.5% on CLIPim and -19.2% on DINO of UltraEdit). This underscores **the limitations of existing benchmarks in evaluating multi-scene editing**. ***(iii)*** In contrast, AnyEdit significantly outperforms SOTA methods across all editing categories, demonstrating its scalability and robustness in handling complex tasks across diverse scenarios. ***(iv)*** Traditional methods often struggle to handle visual editing effectively due to additional visual inputs. In such cases, even

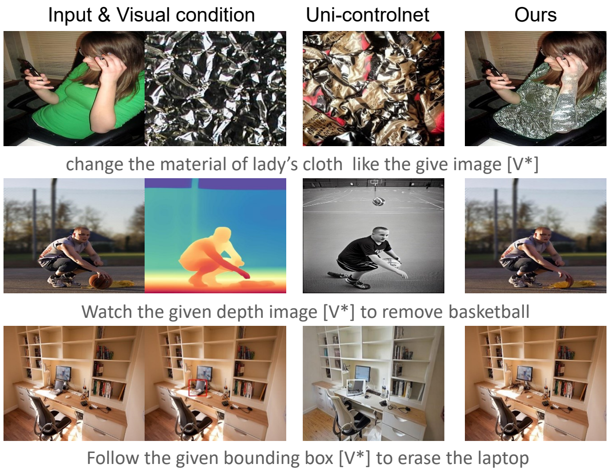


Figure 6. Comparison of our image editing method against Uni-controlnet [58] for visual editing tasks on AnyEdit-Test.

when compared to Uni-ControlNet, which is pre-trained with diverse visual conditions, AnyEdit consistently performs better in visual editing tasks. It shows the efficacy of AnyEdit in handling vision-conditioned editing instructions. We provide more qualitative evidence in Sec. 5.4.

5.4. Qualitative Evaluation

Due to the limitations of quantitative metrics in evaluating editing tasks, we perform qualitative evaluations to further assess the effectiveness of our approach, as shown in Figure 5. Our key observations are: 1) Most baseline models, including the human-tuned HIVE-c[56] and the SOTA UltraEdit with its extensive training data[57], **still suffer from over-editing or misalignment when handling complex fine-grained instructions** (e.g., "facial distortion" and "missing glasses" in Fig. 5(ii)). 2) Due to the limited diversity and quality of current datasets, previous methods (*i.e.*, ip2p [5], MagicBrush [54], and UltraEdit) **struggle to generalize to novel editing types in diverse scenarios** (*e.g.*, failing to follow instructions in rotation change and counting tasks, or roughly altering objects instead of its fine-grained appearance in appearance alter task). 3) In contrast, our method can effectively **ensure editing accuracy in target regions and maintain consistency in irrelevant areas** even without any mask guidance (Fig. 5(i), (vii)). Also, our method can **automatically distinguish between foreground and background** to modify the background (Fig. 5(v)). Moreover, our method **successfully executes more complex editing instructions** (*e.g.*, style change in Fig. 5(vi) and outpainting in Fig. 5(x)).

Additionally, we visualize AnyEdit results on visual editing in Figure 6. In this challenging setting, UniControl can either reflect only the pixel information from the visual condition or retain the semantics of the original image without performing any edits. In contrast, for various visual instructions, AnyEdit **consistently comprehends the pixel information in visual conditions and achieves reliable editing**.

		CLIPim ↑	CLIPout ↑	L1 ↓	DINO ↑
	AnySD w/ AnyEdit	**0.872**	**0.285**	**0.070**	**0.821**
1	w/o task-aware routing	0.838	0.275	0.154	0.757
2	w/o task emb	0.859	0.282	0.107	0.809
3	AnyEdit (w/o compsn.)	0.868	0.271	0.099	0.785

Table 6. Ablation study of our method on EMU-Edit Test [41].

These promising visualization results confirm the effectiveness and high quality of the diverse editing data in AnyEdit. More qualitative results are shown in Appendix H.2.

5.5. In-depth Analysis

AnySD Architecture. We investigate the effectiveness of each component and conduct the following experiments on EMU-Edit Test benchmark: (1) We remove the text-aware routing strategy in AnySD (c.f. Rows 1 of Tab. 6) and find that it leads to significant performance degradation (0.838 v.s. 0.872 in CLIPim and 0.154 v.s. 0.070 in L1), demonstrating its crucial role for adapting diverse image editing tasks. (2) We remove task embeddings in AnySD and observe that it has little impact on semantic alignment but significantly affects visual consistency (c.f. Rows 2 of Tab. 6), suggesting that task embeddings control the perceptual granularity of pixel information during cross-attention.

Analysis of Data Scaling in AnyEdit. In Figure 4, we analyze the data scaling effect of AnyEdit on image editing capability. We observe: (1) For consistency metrics (*i.e.*, CLIPim and DINO), performance improves progressively as data scale increases; (2) For editing accuracy metrics (*i.e.*, CLIPout), we can achieve promising performance even with a small amount of data, indicating that AnyEdit excels in semantic alignment. (3) Furthermore, we remove the AnyEdit-Composition editing data in counterfactual synthesis scenarios, as shown in row 3 of Tab. 6, the lack of conceptual balance in AnySD hinders its ability to generalize, leading to a decline in semantic performance (-4.9% of CLIPout). This confirms the efficacy of counterfactual synthetic scenes to generalization in editing tasks.

6. Conclusion

In this work, we present a novel perspective for categorizing editing tasks and introduce a unified framework that exploits adaptive pipelines to construct high-quality data for diverse editing tasks in a low-resource manner. Building on this, we propose AnyEdit, a multi-type, multi-scene instruction-based editing dataset comprising 2.5M editing samples across 25 distinct types, along with its benchmark, AnyEdit-Test, enabling a more comprehensive paradigm for unified image editing. Furthermore, we develop the powerful AnySD, unlocking the full potential of AnyEdit. Extensive experiments on standard benchmarks and the challenging AnyEdit-Test demonstrate that our method excels in high-quality image editing across diverse tasks and scenarios, accurately executing complex instructions while preserving image consistency in unmodified elements.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (62436007), the Key R&D Projects in Zhejiang Province (No. 2024C01106, 2025C01030), and the Zhejiang NSF (LRG25F020001). We thank all the reviewers for their valuable comments.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 3

[2] Jinbin Bai, Tian Ye, Wei Chow, Enxin Song, Qing-Guo Chen, Xiangtai Li, Zhen Dong, Lei Zhu, and Shuicheng Yan. Meissonic: Revitalizing masked generative transformers for efficient high-resolution text-to-image synthesis. *arXiv preprint arXiv:2410.08261*, 2024. 1

[3] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023. 1, 3

[4] Nitzan Bitton-Guetta, Yonatan Bitton, Jack Hessel, Ludwig Schmidt, Yuval Elovici, Gabriel Stanovsky, and Roy Schwartz. Breaking common sense: Whoops! a vision-and-language benchmark of synthetic and compositional images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2616–2627, 2023. 1

[5] 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, 3, 5, 6, 7, 8

[6] 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*, pages 22560–22570, 2023. 1

[7] Jianbo Chen, Yelong Shen, Jianfeng Gao, Jingjing Liu, and Xiaodong Liu. Language-based image editing with recurrent attentive models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8721–8729, 2018. 1

[8] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6593–6602, 2024. 1

[9] Wei Chow, Juncheng Li, Qifan Yu, Kaihang Pan, Hao Fei, Zhiqi Ge, Shuai Yang, Siliang Tang, Hanwang Zhang, and Qianru Sun. Unified generative and discriminative training for multi-modal large language models. *arXiv preprint arXiv:2411.00304*, 2024. 3

[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 *European Conference on Computer Vision*, pages 88–105. Springer, 2022. 1

[11] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. 1

[12] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 4

[13] Felix Friedrich, Manuel Brack, Lukas Struppek, Dominik Hintersdorf, Patrick Schramowski, Sasha Luccioni, and Kristian Kersting. Fair diffusion: Instructing text-to-image generation models on fairness. *arXiv preprint arXiv:2302.10893*, 2023. 1, 4

[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. *arXiv preprint arXiv:2309.17102*, 2023. 1, 6

[15] Yuying Ge, Sijie Zhao, Chen Li, Yixiao Ge, and Ying Shan. Seed-data-edit technical report: A hybrid dataset for instructional image editing. *arXiv preprint arXiv:2405.04007*, 2024. 1, 3, 4

[16] Yuying Ge, Sijie Zhao, Jinguo Zhu, Yixiao Ge, Kun Yi, Lin Song, Chen Li, Xiaohan Ding, and Ying Shan. Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396*, 2024. 3

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

[18] Rashida Hasan and Cheehung Chu. Noise in datasets: What are the impacts on classification performance?[noise in datasets: What are the impacts on classification performance?]. In *Proceedings of the 11th International Conference on Pattern Recognition Applications and Methods*, 2022. 2

[19] 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, 3, 4

[20] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 6

[21] Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. Hq-edit: A high-quality dataset for instruction-based image editing. *arXiv preprint arXiv:2404.09990*, 2024. 3

[22] Juncheng Li, Xin He, Longhui Wei, Long Qian, Linchao Zhu, Lingxi Xie, Yueting Zhuang, Qi Tian, and Siliang Tang. Fine-grained semantically aligned vision-language pre-training. *Advances in neural information processing systems*, 35:7290–7303, 2022. 3

[23] Juncheng Li, Kaihang Pan, Zhiqi Ge, Minghe Gao, Wei Ji, Wenqiao Zhang, Tat-Seng Chua, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Fine-tuning multimodal

llms to follow zero-shot demonstrative instructions. In *The Twelfth International Conference on Learning Representations*, 2023.

[24] Juncheng Li, Siliang Tang, Linchao Zhu, Wenqiao Zhang, Yi Yang, Tat-Seng Chua, Fei Wu, and Yueting Zhuang. Variational cross-graph reasoning and adaptive structured semantics learning for compositional temporal grounding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12601–12617, 2023.

[25] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26689–26699, 2024. 3, 4

[26] Tsung-Yi Lin, Michael Maire, Serge 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 13*, pages 740–755. Springer, 2014. 1, 3, 4

[27] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. *Advances in Neural Information Processing Systems*, 34:16331–16345, 2021. 1

[28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 3, 4

[29] Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42: 275–293, 2014. 5

[30] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 6, 7

[31] Ranjita Naik and Besmira Nushi. Social biases through the text-to-image generation lens. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 786–808, 2023. 1, 4

[32] Alex 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. *arXiv preprint arXiv:2112.10741*, 2021. 1

[33] Kaihang Pan, Juncheng Li, Hongye Song, Jun Lin, Xiaozhong Liu, and Siliang Tang. Self-supervised meta-prompt learning with meta-gradient regularization for few-shot generalization. *arXiv preprint arXiv:2303.12314*, 2023. 3

[34] Kaihang Pan, Zhaoyu Fan, Juncheng Li, Qifan Yu, Hao Fei, Siliang Tang, Richang Hong, Hanwang Zhang, and Qianru Sun. Towards unified multimodal editing with enhanced knowledge collaboration. *arXiv preprint arXiv:2409.19872*, 2024.

[35] Kaihang Pan, Juncheng Li, Wenjie Wang, Hao Fei, Hongye Song, Wei Ji, Jun Lin, Xiaozhong Liu, Tat-Seng Chua, and Siliang Tang. I3: I ntent-i ntrospective retrieval conditioned on i nstructions. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1839–1849, 2024. 3

[36] Kaihang Pan, Siliang Tang, Juncheng Li, Zhaoyu Fan, Wei Chow, Shuicheng Yan, Tat-Seng Chua, Yueting Zhuang, and Hanwang Zhang. Auto-encoding morph-tokens for multimodal llm. *arXiv preprint arXiv:2405.01926*, 2024. 3

[37] 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

[38] 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. 6

[39] 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

[40] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 1, 2

[41] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8871–8879, 2024. 1, 3, 4, 5, 6, 7, 8

[42] Kihyuk Sohn, Nataniel Ruiz, Kimin Lee, Daniel Castro Chin, Irina Blok, Huiwen Chang, Jarred Barber, Lu Jiang, Glenn Entis, Yuanzhen Li, et al. Styledrop: Text-to-image generation in any style. *arXiv preprint arXiv:2306.00983*, 2023. 1

[43] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248, 2022. 2

[44] 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*, pages 1921–1930, 2023. 6, 7

[45] Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2096–2105, 2023. 2

[46] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Shuting Wang, Tiejun Huang, and

Zheng Liu. Omnigen: Unified image generation. *arXiv preprint arXiv:2409.11340*, 2024. 3

[47] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18381–18391, 2023. 1

[48] Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and CUI Bin. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms. In *Forty-first International Conference on Machine Learning*, 2024. 1

[49] Ling Yang, Bohan Zeng, Jiaming Liu, Hong Li, Minghao Xu, Wentao Zhang, and Shuicheng Yan. Editworld: Simulating world dynamics for instruction-following image editing. *arXiv preprint arXiv:2405.14785*, 2024. 1, 3

[50] Qifan Yu, Juncheng Li, Yu Wu, Siliang Tang, Wei Ji, and Yueting Zhuang. Visually-prompted language model for fine-grained scene graph generation in an open world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21560–21571, 2023. 1

[51] Qifan Yu, Juncheng Li, Wentao Ye, Siliang Tang, and Yueting Zhuang. Interactive data synthesis for systematic vision adaptation via llms-aigcs collaboration. *arXiv preprint arXiv:2305.12799*, 2023. 1

[52] Qifan Yu, Juncheng Li, Longhui Wei, Liang Pang, Wentao Ye, Bosheng Qin, Siliang Tang, Qi Tian, and Yueting Zhuang. Hallucidoctor: Mitigating hallucinatory toxicity in visual instruction data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12944–12953, 2024. 3

[53] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9150–9161, 2023. 1, 3, 4

[54] 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, 2024. 1, 3, 5, 6, 7, 8

[55] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 1

[56] Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, Caiming Xiong, and Ran Xu. Hive: Harnessing human feedback for instructional visual editing. *arXiv preprint arXiv:2303.09618*, 2023. 6, 7, 8

[57] Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. *arXiv preprint arXiv:2407.05282*, 2024. 1, 3, 4, 5, 6, 7, 8

[58] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K. Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 2023. 6, 7, 8

[59] Dewei Zhou, You Li, Fan Ma, Xiaoting Zhang, and Yi Yang. Migc: Multi-instance generation controller for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6818–6828, 2024. 1

[60] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022. 6

[61] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 1