Beyond Simple Edits: X-Planner for Complex Instruction-Based Image Editing

Chun-Hsiao Yeh[1,3], **Yilin Wang**[3], **Nanxuan Zhao**[3], **Richard Zhang**[3], **Yuheng Li**[3], **Yi Ma**[1,2], **Krishna Kumar Singh**[3]

[1]UC Berkeley
[2]HKU
[3]Adobe Research

Abstract

Recent diffusion-based image editing methods have made great strides in text-guided tasks but often struggle with complex, indirect instructions. Additionally, current models frequently exhibit poor identity preservation, unintended edits, or rely on manual masks. To overcome these limitations, we introduce **X-Planner**, a Multimodal Large Language Model (MLLM)-based planning system that bridges user intent with editing model capabilities. **X-Planner** uses chain-of-thought reasoning to systematically break down complex instructions into simpler sub-instructions. For each one, **X-Planner** automatically generates precise edit types and segmentation masks, enabling localized, identity-preserving edits without applying external tools or models during inference. To enable the training of such a planner, we also introduce a fully automated, reproducible pipeline to generate large-scale, high-quality training data. Our complete system achieves state-of-the-art results on both existing and newly proposed complex instruction-based editing benchmarks.

Introduction

Despite impressive progress in diffusion-based image editing (Podell et al. 2023; Brooks, Holynski, and Efros 2023; Zhang et al. 2024a), existing systems often struggle with a simple but critical challenge: understanding and executing *complex, multi-step user instructions*. While free-form methods (Brooks, Holynski, and Efros 2023; Zhang et al. 2024a; Sheynin et al. 2024) offer convenience but often misinterpret intent, controllable approaches (Nichol et al. 2021; Zhao et al. 2024; Li et al. 2023; Wang et al. 2024a; Shi et al. 2024; Nie et al. 2024; Ye et al. 2023; Chen et al. 2024a) provide spatial precision yet demand tedious manual inputs like masks or bounding boxes.

At the core lies a deeper issue: ***the disconnect between user's instruction and editing models' understanding, reasoning, and execution***. We identify three major challenges that highlight this gap: As illustrated in Figure 1, ***(1) Multi-Object Reasoning:*** Instructions like "make this meal setting to breakfast theme" (row 1) require correctly identifying and modifying multiple targets, not just one. ***(2) Multi-Task Planning:*** Prompts often imply multiple distinct edits (e.g., change style + alter object texture) (row 2,3), demanding coherent decomposition and spatial control. ***(3) Indirect and Implicit Cues:*** Instructions such as "make the image look like the season when ice cream is a daily need" (row 4) require contextual reasoning and visual transformations beyond literal object edits.

Recent MLLM-guided methods (Huang et al. 2024; Fu et al. 2023) often falter on such nuanced instructions, misinterpreting prompts or editing irrelevant regions. While GenArtist (Wang et al. 2024b) takes a step toward instruction decomposition using GPT-4 (Achiam et al. 2023), it relies on brittle external detectors (Liu et al. 2024; Kirillov et al. 2023), causing failure in tasks like object insertion where the target is absent from the image and must be hallucinated (e.g., "add a red ball" in Figure 1).

In order to robustly handle complex editing instructions, we need an unified system that *reasons over image content, understands instruction structure, and generates spatially grounded, edit-type-aware plans*—all within a single, unified model. To achieve this goal, we propose ***X-Planner***, an Multimodal Large Language Model (MLLM)-driven editing agent that interprets complex prompts, decomposes them into actionable sub-instructions, and generates precise masks or bounding boxes tailored to each edit type. For example, for insertion tasks, it goes beyond masking and predicts a plausible insertion region using world knowledge—something external detectors cannot do. *X-Planner* also outputs the edit type, allowing task-specific model selection to improve quality and control.

To support this system, we introduce COMPIE, a large-scale dataset of 260K automatically generated complex-to-simple instruction pairs, each annotated with masks, edit types, and insertion boxes. This dataset is created with a fully automated pipeline and strict quality controls. In light of the lack of benchmarks for evaluating complex instruction-driven image editing, we also propose a comprehensive evaluation protocol and a benchmark on complex instructions. Our contributions are summarized as follows:

- **An unified and effective planning agent.** We introduce *X-Planner*, an MLLM-driven agent that automatically decomposes complex instructions into simpler tasks and generates precise spatial guidance (masks and boxes), internalizing independently of external tools like detectors, segmentors, or external models at inference time.

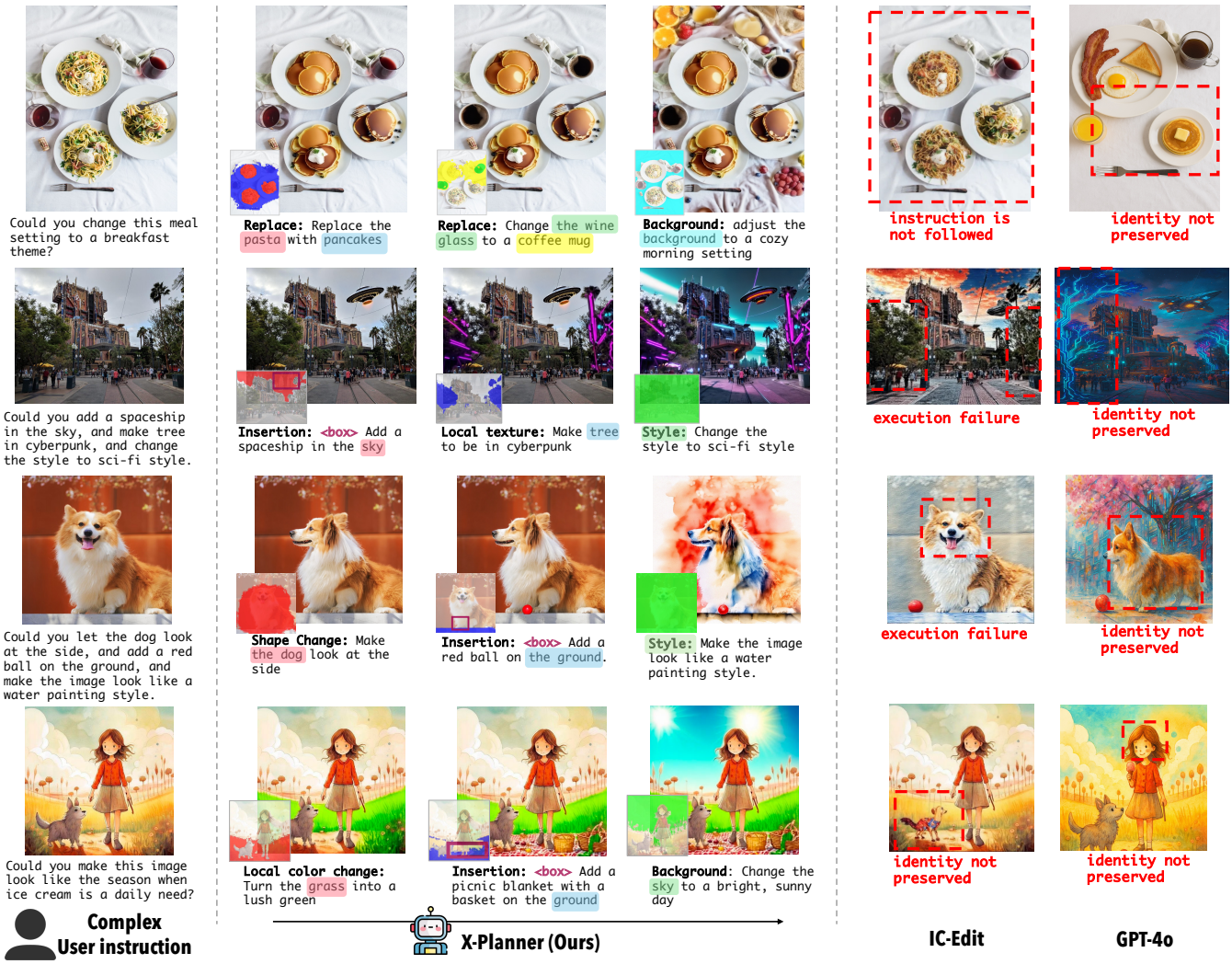

Figure 1: **Left.** Given a source image and complex instruction, *X-Planner* decomposes it into sub-instructions with edit types, generates corresponding segmentation masks, and predicts bounding boxes for insertion edits. Each edit is executed by passing the sub-instruction and region (mask/box) to a compatible editing model. **Right.** Recent IC-Edit (Zhang et al. 2025) and GPT-4o (Hurst et al. 2024) struggle with complex instruction understanding and object identity preservation.

- **A fully automated and reproducible data pipeline.** We present a novel, large-scale data creation pipeline that systematically generates complex-simple instruction pairs with corresponding annotations (segmentation masks, bounding boxes, and edit types).
- **A new complex editing benchmark with extensive validation.** We introduce a new benchmark, COMPIE, and show through comprehensive experiments that our framework is not only superior to prior art but also robust against multi-step errors via a closed-loop verification mechanism.

Related Works

Controllable Generative Image Editing. Recent text-to-image diffusion models (Rombach et al. 2022; Podell et al. 2023) have inspired text-guided image editing. These approaches are often training-free (Hertz et al. 2022; Parmar et al. 2023; Wu et al. 2024), like Prompt-to-Prompt (Hertz et al. 2022), which steer the diffusion process, or training-based (Fu et al. 2023; Ge et al. 2024), like InstructPix2Pix (Brooks, Holynski, and Efros 2023), which fine-tune on paired images. Despite their effectiveness, these methods can over-edit or misalign with user intent. Subsequent works (Zhao et al. 2024; Chen, Laina, and Vedaldi 2024; Shi et al. 2024; Ye et al. 2023; Chen et al. 2024a) introduced control signals—such as masks, boxes, or dragging—to improve precision. However, these controls rely on manual inputs and are limited to simple, direct instructions. Our *X-Planner* addresses these limitations. It automatically decomposes complex user instructions into actionable sub-instructions and generates spatial guidance (segmentation

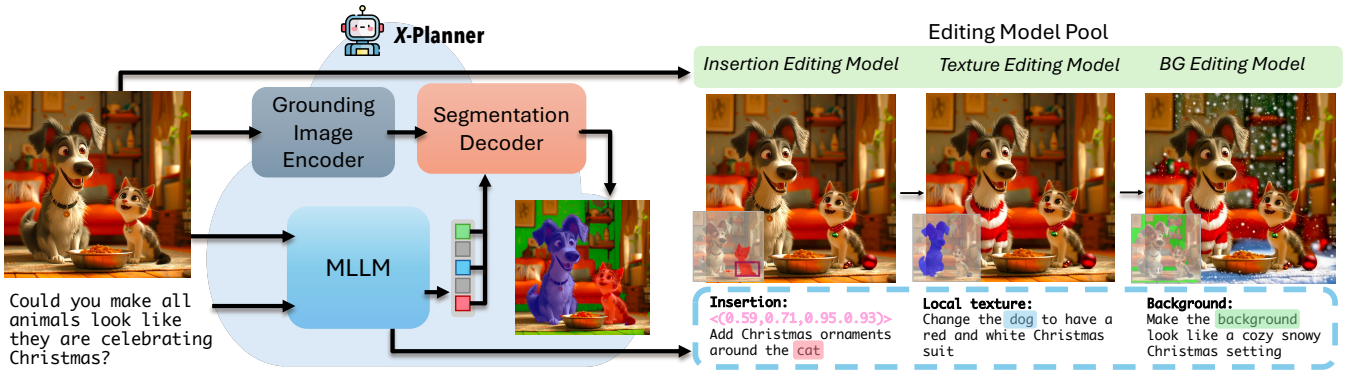

Figure 2: **Overview of *X-Planner* for Complex Instruction-Based Editing.** *X-Planner* first uses an MLLM to decompose complex instructions into sub-instructions with editing anchors, which are fed to a segmentation decoder for mask generation. For insertion edits, it also predicts bounding boxes. Each sub-instruction is then executed by selecting the appropriate editing model from a model pool to execute each specific edit given *X-Planner* generated sub-instruction along with masks / boxes.

masks and bounding boxes), removing the need for manual controls and enabling fine-grained, intent-aligned edits.

MLLM-based Image Editing. Multimodal Large Language Models (MLLMs) (Chowdhery et al. 2023; Touvron et al. 2023; Dubey et al. 2024) are increasingly used for image editing. Methods like SmartEdit (Huang et al. 2024) and MGIE (Fu et al. 2023) use MLLMs to guide editing models but can struggle with identity preservation. GenArtist (Wang et al. 2024b) uses closed-source GPT-4 to decompose prompts and relies on external detectors for spatial grounding. In contrast, our *X-Planner* is a unified MLLM agent that handles complex instructions by decomposing them into actionable steps. It uniquely generates edit-type-specific masks—even hallucinating regions for object insertion. Unlike prior work, *X-Planner* requires no external tools or closed-source models.

Method

We introduce *X-Planner*, a unified planning agent for complex instruction-based image editing. Unlike models relying on handcrafted inputs or brittle toolchains, *X-Planner* directly predicts actionable sub-instructions and control guidance from complex prompts to enable precise, task-type-specific editing (Figure 2).

From Limitations to Design: Why *X-Planner*?

Prior work like GLaMM (Rasheed et al. 2024) can ground visual entities but is not designed for instruction-driven editing. We observe two key limitations: **(1) Lack of instruction understanding:** They are trained for grounding, not language decomposition, failing to parse complex prompts (e.g., *"make the tree cyberpunk and change the style to sci-fi,"*). **(2) Inability to generate control inputs:** The segmentation decoder only works for visible objects. For insertion tasks (e.g., "add a red ball" in Figure 1), GLaMM cannot localize hallucinated objects or predict insertion regions.

These failures reveal a lack of structured, multi-step planning. *We argue that a decomposition-based approach offers fundamental advantages in interpretability, controllability, and modular efficiency over a monolithic model.* It allows users or verification systems to correct single-step errors and enables dispatching sub-tasks to specialized editing models.

Guided by this, we re-purpose the GLaMM architecture for editing-centric planning. *X-Planner* takes an image and complex instruction, then outputs a sequence of decomposed sub-instructions, each annotated with: *Edit type*, *Editing anchor* (target object/region), and *Control inputs* (a segmentation mask and, for insertion, a bounding box, e.g., *[insertion]; 0.59,0.71,0.95,0.93> Add Christmas ornaments around the cat*). We retain the GLaMM vision-segmentation backbone but retrain its MLLM on our proposed COMPIE dataset. This enables *X-Planner* to understand complex instructions and generate grounded, task-aware control inputs, including plausible insertion boxes, bypassing external detectors.

Since no existing dataset supports such end-to-end planning, we construct COMPIE, a large-scale dataset of 260K examples linking complex instructions to simplified steps, annotated with masks, edit types, and insertion boxes. This supervision enables *X-Planner* to learn instruction decomposition and spatial control jointly, making it the first fully-trainable, end-to-end planner for multimodal image editing.

Automated Data Annotation Pipeline

To support *X-Planner*'s training, we construct **COMPIE**, a novel, large-scale dataset tailored for complex instruction decomposition and spatial control. Our automated pipeline comprises three levels: **Level 1** (Figure 3) uses MLLM to generate complex-to-simple instruction pairs with editing anchors; **Level 2** (Figure 4) employs Grounded-SAM (Ren et al. 2024) to produce and refine segmentation masks based on edit type; **Level 3** (Figure 5) predicts insertion boxes for hallucinated objects, enabling location-aware insertion beyond visible object segmentation.

Level 1: Complex-Simple Instruction Pair Generation. Current editing models lack data for indirect or multi-step instructions. To address this, we leverage GPT-4o (Hurst et al. 2024) and Pixtral-Large (Agrawal et al. 2024) to gen-

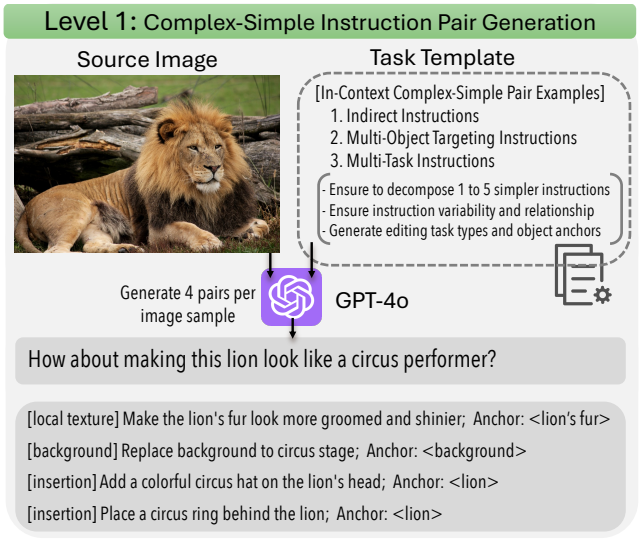

Figure 3: **Level 1: Complex-Simple Instruction Pair Generation.** Using our structured template, we prompt GPT-4o to generate complex instructions—including indirect, multi-object, and multi-task instructions (as defined in Section 1)—along with their corresponding simpler instructions, object anchors, and edit types.

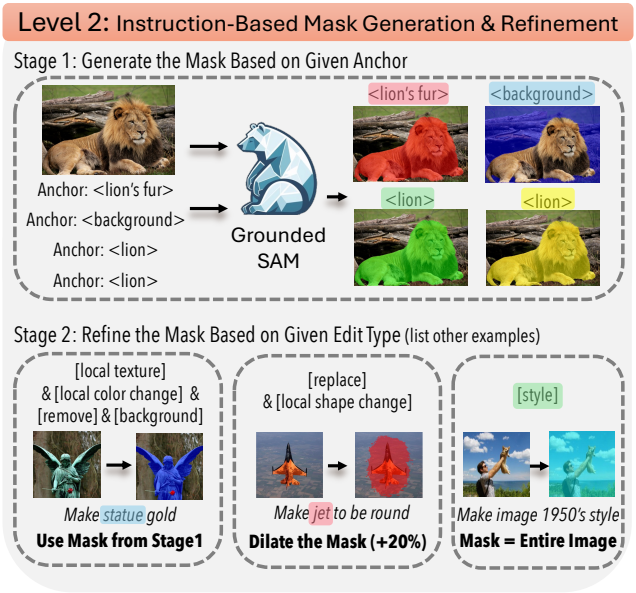

Figure 4: **Level 2: Instruction-Based Mask Generation and Refinement.** Stage 1 uses the source image and anchor text with Grounded-SAM to generate an initial object mask. Stage 2 refines the mask using edit-type-specific strategies defined in Level 1 (Figure 3).

erate *complex-to-simple* instruction pairs grounded in image content from diverse sources (SEED-X (Ge et al. 2024), UltraEdit (Zhao et al. 2024), InstructPix2Pix (Brooks, Holynski, and Efros 2023)) into atomic edits tagged with *edit types* and *editing anchor(s)*. By utilizing in-context exemplars for various reasoning patterns (e.g., indirect, multi-task) and injecting *simple-simple* pairs, we ensure robust generalization across 1 to 5 atomic steps. Please see Supp. for more details.

Level 2: Instruction Mask Generation and Refinement. We define target regions in two stages. First (Stage 1), we use Grounded-SAM (Ren et al. 2024) to generate raw masks from Level 1 anchors. Then (Stage 2), we refine these based on edit type: using direct masks for texture or background changes, and a 20% dilated mask for shape edits. For *replace* tasks (e.g., *"replace pasta with pancake"*), we use the union of pre- and post-edit masks (if available) or dilate the pre-edit mask by 20%, while global style edits utilize the entire image (Figure 4).

Level 3: Insertion Task-Based Box Localization. A key shortcoming of existing pipelines is handling insertion prompts—where the object does not yet exist. Segmenting the anchor (e.g., "lion" in *"add a circus ring behind the lion"* in Figure 3) misrepresents the intended location and causes errors by segmenting the existing object (lion) rather than the intended insertion area (behind the lion). To solve this, we fine-tune the MLLM from GLaMM (Rasheed et al. 2024) to predict plausible bounding boxes for unseen objects. We supervise this on the MULAN dataset (Tudosiu et al. 2024), which includes background images with and without foreground objects, allowing us to use images without the object as input and MLLM learns to predict the

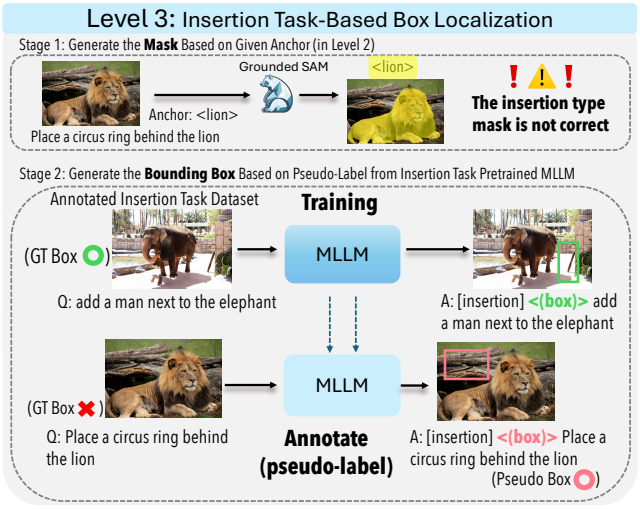

Figure 5: **Level 3: Insertion Task-Based Mask & Box Localization.** Grounded-SAM cannot localize objects absent from the source image. We pre-train an MLLM on box-annotated data (Tudosiu et al. 2024) to predict bounding boxes for insertion edits via pseudo-labeling.

bounding box for novel insertion instructions. For unannotated examples, the model produces pseudo-labels from the instruction alone, enabling large-scale generation. These predictions are diverse yet consistent (see Figure 7), without relying on external detectors for insertion instructions.

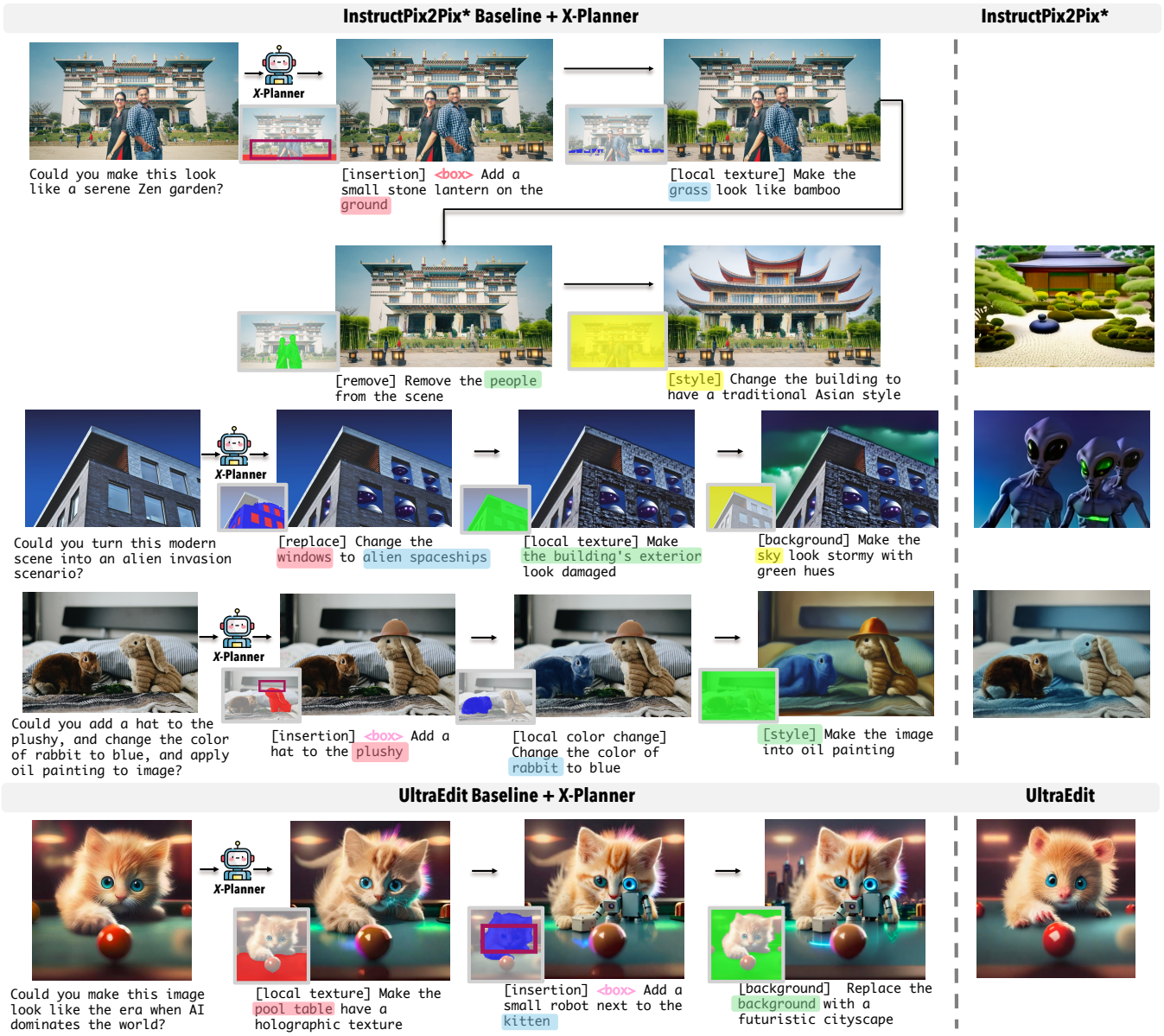

Figure 6: **Qualitative Comparison on Complex Instruction Editing.** Examples illustrating how *X-Planner* can improve identity preservation and instruction alignment for baselines like InstructPix2Pix* and UltraEdit. By providing decomposed steps with spatial guidance (masks/boxes), our planner facilitates more precise edits than methods relying only on the global prompt.

Experiments

We evaluate *X-Planner* on both simple and complex instruction settings. For simple edits, we use both the MagicBrush (Zhang et al. 2024a) and Emu Edit (Sheynin et al. 2024) benchmarks. For complex edits, we assess performance w/wo *X-Planner* using our COMPIE benchmark.

Settings. *X-Planner* is built on GLaMM (Rasheed et al. 2024) with a Vicuna-7B backbone (Zheng et al. 2023), trained on 260K complex-simple pairs from COMPIE. See Supp. for details of training and dataset distributions.

Baselines. For MagicBrush and Emu Edit, we compare against existing methods (Zhang et al. 2024b,a; Meng et al. 2021; Zhao et al. 2024; Xiao et al. 2024), and integrate *X-Planner* into UltraEdit (Zhao et al. 2024) to assess its benefit under simple settings. For COMPIE, we benchmark against UltraEdit and InstructPix2Pix*, an improved version of InstructPix2Pix using our internal dataset and also utilizes mask as input conditioning, to show the generalizability of *X-Planner*. We also include MLLM-based methods MGIE (Fu et al. 2023), SmartEdit (Huang et al. 2024), and GenArtist (Wang et al. 2024b) for comparison. Each baseline is evaluated with and without *X-Planner*.

Metrics. For MagicBrush and Emu Edit, we follow its standard setup using L1/L2 distance, CLIP-I, and DINO sim-

ilarity. For COMPIE, we adopt EmuEdit (Sheynin et al. 2024)'s protocol: L1, $CLIP_{im}$, DINO for content preservation; Given that $CLIP_{out}$ can struggle to capture the nuances of complex instructions, we additionally employ InternVL2-Llama3-76B (Chen et al. 2024b), to evaluate the alignment between the editing instruction and edited image ($MLLM_{ti}$); and $MLLM_{im}$ for image similarity. We use InternVL2-76B for evaluation to avoid overlap with GPT-4o used in training. Please find Supp. for more details.

	Method	Control	L1↓	L2↓	CLIP-I↑	DINO↑
MagicBrush	*— Single-Turn —*					
	Null Text Inversion	—	0.0749	0.0197	0.8827	0.8206
	HIVE	—	0.1092	0.0380	0.8519	0.7500
	InstructPix2Pix (IP2P)	—	0.1141	0.0371	0.8512	0.7437
	IP2P w/ MagicBrush	—	0.0625	0.0203	0.9332	0.8987
	UltraEdit	—	0.0614	0.0181	0.9197	0.8804
	UltraEdit	Human Mask	0.0575	0.0172	0.9307	**0.8982**
	X-Planner + UltraEdit	Mask	0.0528	0.0171	0.9281	0.8900
	X-Planner + UltraEdit	Mask + Bbox	0.0513	**0.0168**	0.9312	0.8959
	X-Planner + Bag of Models	Mask + Bbox	**0.0511**	0.0172	**0.9331**	0.8970
	— Multi-Turn —					
	Null Text Inversion	—	0.1057	0.0335	0.8468	0.7529
	HIVE	—	0.1521	0.0557	0.8004	0.6463
	InstructPix2Pix (IP2P)	—	0.1345	0.0460	0.8304	0.7018
	IP2P w/ MagicBrush	—	0.0964	0.0353	0.8924	0.8273
	UltraEdit	—	0.0780	0.0246	0.8954	0.8322
	UltraEdit	Human Mask	0.0745	0.0236	0.9045	0.8505
	X-Planner + UltraEdit	Mask	0.0679	0.0227	0.9025	0.8423
	X-Planner + UltraEdit	Mask + Bbox	0.0668	0.0226	0.9047	0.8475
	X-Planner + Bag of Models	Mask + Bbox	**0.0665**	**0.0223**	**0.9079**	**0.8508**
Emu Edit	InstructPix2Pix (450K)	—	0.1213	-	0.8518	0.7656
	OmniGen	—	-	-	0.8360	0.8040
	EmuEdit (10M)	—	0.0895	-	0.8622	0.8358
	UltraEdit (1M w/o region)	—	0.0515	-	0.8915	0.8656
	UltraEdit (3M w/o region)	—	0.0713	-	0.8446	0.7937
	UltraEdit	—	0.0611	-	0.8627	0.8079
	X-Planner + UltraEdit	Mask	0.0462	-	0.9007	0.8723
	X-Planner + UltraEdit	Mask + Bbox	0.0457	-	0.9029	**0.8766**
	X-Planner + Bag of Models	Mask + Bbox	**0.0443**	-	**0.9046**	0.8754

Table 1: **Quantitative Comparison on Simple Instruction-Based Benchmarks.** We report results for MagicBrush single- and multi-turn settings, and Emu Edit benchmark. In comparison to UltraEdit baseline using human labeled masks, our *X-Planner* uses its predicted masks and bounding boxes as control inputs. *For Bag of Models, we utilize PowerPaint for removal tasks, InstructDiff for style changes, and UltraEdit for other edit types.*

Results on Simple Instruction-Based Benchmarks

In Table 1, we show quantitative results on the MagicBrush (Zhang et al. 2024a), and Emu Edit (Sheynin et al. 2024) benchmarks. Key observations: (1) *X-Planner* enhances UltraEdit by providing masks and bounding boxes for localized edits, improving performance, especially for insertion tasks. *X-Planner*'s mask is able to match the human labeled mask, which serve as a practical upper bound, and even outperform it in most of metrics as shown in the result. (2) *X-Planner* is model-agnostic, integrating seamlessly with multiple models (e.g., PowerPaint (Zhuang et al. 2023) for removal, InstructDiff (Geng et al. 2024) for style changes, and UltraEdit for other edits)for boosting overall performance. Please find Supp. for more quantitative results.

Results on Complex Instruction-Based Benchmark

Qualitative Results. Figure 6 shows editing results w/wo *X-Planner* for InstructPix2Pix* and UltraEdit. Without *X-Planner*, models often misinterpret complex instructions (e.g., UltraEdit fails to capture the "futuristic" style in the last row). Even when the instruction is understood, baselines struggle with identity preservation due to lack of spatial guidance (e.g., first row for InstructPix2Pix*), whereas *X-Planner*'s controls improve consistency and localization.

Quantitative Results. We construct COMPIE-Eval, a 550-image test set focused on complex edits, sourced from LAION-high-aesthetics (Schuhmann et al. 2022) and Unsplash-2K (Kim and Son 2021). Instructions are generated using GPT-4o and filtered by crowd workers. The dataset includes instruction types defined in Section 1.

As shown in Table 2, *X-Planner* significantly improves editing models by decomposing instructions and providing precise masks, boosting the identity preservation. Gains are consistent across most metrics for UltraEdit and InstructPix2Pix*, except $CLIP_{out}$, which sometimes misaligns image and complex prompt semantics. To better capture instruction alignment, we use $MLLM_{ti}$ (InternVL2 (Chen et al. 2024b)), which shows substantial improvements when using *X-Planner*. MLLM-guided baselines MGIE and SmartEdit perform worse in both structure and alignment. GenArtist (Wang et al. 2024b) uses external segmentors could not handle several edit types, especially insertion.

Reproducibility with Open-Source Models. To ensure our pipeline is not reliant on closed-source models, we trained *X-Planner* on data generated by using the open-source MLLM, Pixtral-Large (Agrawal et al. 2024). As shown in Table 2, *X-Planner* trained on this version achieves performance comparable to the GPT-4o-based version.

User Study. We conducted a user study with 100 random samples from 550 images in the COMPIE benchmark to compare results of two baselines: InstructPix2Pix* and UltraEdit with and without *X-Planner*. 10 Participants rated images on (1) identity preservation, (2) instruction alignment, and (3) overall quality, and choose the preferred image or rate them equal. In Figure 8, we show average results for both benchmarks and observe the users prefer results with *X-Planner* for all criteria (better means *X-Planner* preferred).

Ablation Studies and Robustness Analysis

Control Input (Mask & Bbox) Generation. We first evaluate the quality of the spatial guidance generated by *X-Planner*. As shown in Table 3, for *X-Planner*'s segmentation mask generation performance on the PIE benchmark (Ju et al. 2024), comparing it with GLaMM (Rasheed et al. 2024) variations (baseline model and a version fine-tuned on RefSeg dataset to have better grounding) and a baseline that leverages Llama3 (Dubey et al. 2024) for object anchoring followed by GLaMM for mask generation. Based on the results, we can see *X-Planner* consistently surpasses other instruction-to-segmentation methods. For bounding box generation in insertion tasks, Table 4 shows a detailed ablation on the MULAN benchmark (Tudosiu et al. 2024). Our proposed techniques, including pseudo-labeling and

Backbone	Method	Control	L1↓	CLIP$_{im}$↑	CLIP$_{out}$↑	DINO↑	MLLM$_{ti}$↑	MLLM$_{im}$↑
	SmartEdit (Huang et al. 2024)	—	0.2764	0.7713	0.2512	0.6044	0.6511	0.5347
	MGIE (Fu et al. 2023)	—	0.2988	0.7692	0.2498	0.5981	0.6408	0.5288
UltraEdit	UltraEdit (Baseline)	—	0.1292	0.7688	**0.2698**	0.6387	0.6652	0.5523
UltraEdit	GenArtist (Wang et al. 2024b) + UltraEdit	Mask + Bbox	0.1279	0.7704	0.2654	0.6412	0.6664	0.5541
UltraEdit	*X-Planner* (trained w GPT-4o generated data) + UltraEdit	Decomposed Instruction Only	0.1253	0.7767	0.2621	0.6435	0.6894	0.5593
UltraEdit	*X-Planner* (trained w GPT-4o generated data) + UltraEdit	Decomposed Instruction + Mask + Bbox	**0.1188**	**0.7875**	0.2569	**0.6599**	0.7061	0.5744
UltraEdit	*X-Planner* (trained w Pixtral-Large generated data) + UltraEdit	Decomposed Instruction Only	0.1261	0.7744	0.2630	0.6428	0.6904	0.5626
UltraEdit	*X-Planner* (trained w Pixtral-Large generated data) + UltraEdit	Decomposed Instruction + Mask + Bbox	0.1207	0.7853	0.2584	0.6577	**0.7102**	**0.5765**
InstructPix2Pix*	InstructPix2Pix* (Baseline)	—	0.1517	0.8020	**0.2666**	0.6988	0.6727	0.6160
InstructPix2Pix*	GenArtist (Wang et al. 2024b) + InstructPix2Pix*	Mask + Bbox	0.1501	0.8079	0.2653	0.7045	0.6689	0.6131
InstructPix2Pix*	*X-Planner* (trained w GPT-4o generated data) + InstructPix2Pix*	Decomposed Instruction Only	0.1458	0.8143	0.2641	0.7114	0.7072	0.6277
InstructPix2Pix*	*X-Planner* (trained w GPT-4o generated data) + InstructPix2Pix*	Decomposed Instruction + Mask + Bbox	**0.1320**	0.8285	0.2591	0.7068	0.7408	0.6454
InstructPix2Pix*	*X-Planner* (trained w Pixtral-Large generated data) + InstructPix2Pix*	Decomposed Instruction Only	0.1460	0.8141	0.2655	0.7122	0.7088	0.6295
InstructPix2Pix*	*X-Planner* (trained w Pixtral-Large generated data) + InstructPix2Pix*	Decomposed Instruction + Mask + Bbox	0.1325	**0.8291**	0.2586	**0.7077**	**0.7431**	**0.6488**

Table 2: **Master Quantitative Comparison on the COMPIE-Eval Benchmark.** This table provides a comprehensive breakdown for both UltraEdit and InstructPix2Pix* backbones. It shows *X-Planner*'s superiority over baselines and highlights the key contributions of reproducibility (training data via open-source model, Pixtral-Large). To overcome the limitations of $CLIP_{out}$ in handling complex instructions, we utilize an MLLM-based evaluation metric to better reflect *X-Planner*'s capabilities.

box enlargement, systematically boost performance at both K=1 and K=3, with the combination of masks and boxes achieving the best results. Finally, Figure 7 shows our predicted bounding boxes are plausible and consistent.

Method	IoU	Precision	Recall
Rand. 10% Mask	0.09	0.49	0.12
GLaMM-Base	0.14	0.66	0.15
GLaMM-RefSeg	0.28	0.69	0.32
Llama3+GLaMM	0.44	0.73	0.53
X-Planner (Ours)	**0.67**	**0.79**	**0.81**

Table 3: **Mask Generation on PIE Benchmark**.

Setting	K=1		K=3	
	IoU	AP$_{50}$	IoU	AP$_{50}$
Mask Only	0.37	0.34	0.46	0.37
+Pseudo-L	0.63	0.70	0.71	0.69
+Box Enlar.	**0.75**	0.77	**0.81**	0.82
+Mask&Box	0.73	**0.78**	**0.81**	**0.86**

Table 4: **BBox Localization on MULAN Benchmark**.

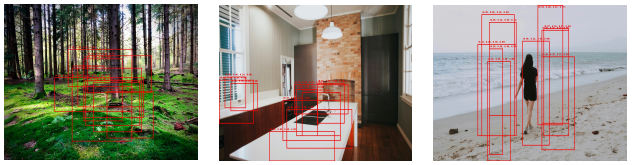

Figure 7: **Visualize Consistent Bounding Box with Repeated Runs.** *X-Planner* generates consistent and plausible bounding boxes for insertion tasks over 10 runs.

Robustness via Closed-Loop Error Verification. To address potential error propagation in multi-step editing, we introduce a closed-loop verification mechanism. After each editing step, a MLLM verifier (e.g., InternVL2.5-38B (Chen et al. 2024b) and GPT-4o (Hurst et al. 2024)) to assign a score from 0–4 that reflects how well the edited image aligns with the current instruction; if the score is below a threshold (e.g., 3), the step is automatically re-generated using a different random seed to recover from potential hallucinations, misalignment (see Appendix for prompt and success rate). Table 5 compares our method against baselines. The verification mechanism provides further improvements; Even a single verification attempt (max=1) boosts performance across all metrics, while allowing more retries (max=4) yields additional gains, especially in $MLLM_{ti}$.

Method	Visual Consistency			Instruction Alignment	
	DINO	CLIP$_{im}$	$MLLM_{im}$	$CLIP_{out}$	$MLLM_{ti}$
SmartEdit	0.6044	0.7713	0.5347	0.2512	0.6511
MGIE	0.5981	0.7692	0.5288	0.2498	0.6408
UltraEdit (UE)	0.6387	0.7688	0.5523	**0.2698**	0.6652
X-Planner+UE (No Verify)	0.6599	0.7875	0.5744	0.2569	0.7061
+ Verify (GPT-4o, max=1)	0.6612	0.7853	0.5798	0.2563	0.7113
+ Verify (InternVL, max=1)	0.6632	0.7861	0.5822	0.2559	0.7128
+ Verify (GPT-4o, max=4)	**0.6673**	**0.7942**	0.5936	0.2574	**0.7308**
+ Verify (InternVL, max=4)	0.6647	0.7901	**0.5955**	0.2571	0.7258

Table 5: **Effectiveness of MLLM-Based Verification.** For COMPIE-Eval, closed-loop verification boosts the *X-Planner*+UltraEdit model against other baselines. We notice more retries (max=4) offers additional gains.

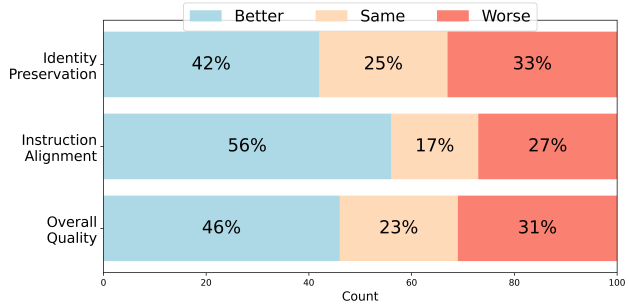

Figure 8: **User Study on COMPIE-Eval Benchmark.** We compare against InstructPix2Pix* and UltraEdit. "Better" means the generated images by using our *X-Planner* is preferred and vice versa.

Conclusion

In this paper, we introduced *X-Planner* that handles complex image editing by decomposing instructions and generating spatial control guidance such as mask and box. By bridging high-level semantic logic with low-level execution, our "decompose, ground, and execute" framework effectively translates abstract user intents into executable atomic tasks with mask and box supervision. This approach mitigates identity inconsistencies in multi-step editing scenarios.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Agrawal, P.; Antoniak, S.; Hanna, E. B.; Bout, B.; Chaplot, D.; Chudnovsky, J.; Costa, D.; De Monicault, B.; Garg, S.; Gervet, T.; et al. 2024. Pixtral 12B. *arXiv preprint arXiv:2410.07073*.

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

Chen, M.; Laina, I.; and Vedaldi, A. 2024. Training-free layout control with cross-attention guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 5343–5353.

Chen, X.; Huang, L.; Liu, Y.; Shen, Y.; Zhao, D.; and Zhao, H. 2024a. Anydoor: Zero-shot object-level image customization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6593–6602.

Chen, Z.; Wang, W.; Tian, H.; Ye, S.; Gao, Z.; Cui, E.; Tong, W.; Hu, K.; Luo, J.; Ma, Z.; et al. 2024b. Internvl2: Better than the best—expanding performance boundaries of open-source multimodal models with the progressive scaling strategy.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Fu, T.-J.; Hu, W.; Du, X.; Wang, W. Y.; Yang, Y.; and Gan, Z. 2023. Guiding instruction-based image editing via multimodal large language models. *arXiv preprint arXiv:2309.17102*.

Ge, Y.; Zhao, S.; Zhu, J.; Ge, Y.; Yi, K.; Song, L.; Li, C.; Ding, X.; and Shan, Y. 2024. Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396*.

Geng, Z.; Yang, B.; Hang, T.; Li, C.; Gu, S.; Zhang, T.; Bao, J.; Zhang, Z.; Li, H.; Hu, H.; et al. 2024. Instructdiffusion: A generalist modeling interface for vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12709–12720.

Hertz, A.; Mokady, R.; Tenenbaum, J.; Aberman, K.; Pritch, Y.; and Cohen-Or, D. 2022. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*.

Huang, Y.; Xie, L.; Wang, X.; Yuan, Z.; Cun, X.; Ge, Y.; Zhou, J.; Dong, C.; Huang, R.; Zhang, R.; et al. 2024. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8362–8371.

Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Ju, X.; Zeng, A.; Bian, Y.; Liu, S.; and Xu, Q. 2024. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. In *The Twelfth International Conference on Learning Representations*.

Kim, Y.; and Son, D. 2021. Noise Conditional Flow Model for Learning the Super-Resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4015–4026.

Li, Y.; Liu, H.; Wu, Q.; Mu, F.; Yang, J.; Gao, J.; Li, C.; and Lee, Y. J. 2023. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22511–22521.

Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Jiang, Q.; Li, C.; Yang, J.; Su, H.; et al. 2024. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, 38–55. Springer.

Meng, C.; He, Y.; Song, Y.; Song, J.; Wu, J.; Zhu, J.-Y.; and Ermon, S. 2021. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*.

Nichol, A.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

Nie, W.; Liu, S.; Mardani, M.; Liu, C.; Eckart, B.; and Vahdat, A. 2024. Compositional Text-to-Image Generation with Dense Blob Representations. *arXiv preprint arXiv:2405.08246*.

Parmar, G.; Kumar Singh, K.; Zhang, R.; Li, Y.; Lu, J.; and Zhu, J.-Y. 2023. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, 1–11.

Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; and Rombach, R. 2023. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*.

Rasheed, H.; Maaz, M.; Shaji, S.; Shaker, A.; Khan, S.; Cholakkal, H.; Anwer, R. M.; Xing, E.; Yang, M.-H.; and Khan, F. S. 2024. Glamm: Pixel grounding large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13009–13018.

Ren, T.; Liu, S.; Zeng, A.; Lin, J.; Li, K.; Cao, H.; Chen, J.; Huang, X.; Chen, Y.; Yan, F.; et al. 2024. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Schuhmann, C.; Beaumont, R.; Vencu, R.; Gordon, C.; Wightman, R.; Cherti, M.; Coombes, T.; Katta, A.; Mullis, C.; Wortsman, M.; et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35: 25278–25294.

Sheynin, S.; Polyak, A.; Singer, U.; Kirstain, Y.; Zohar, A.; Ashual, O.; Parikh, D.; and Taigman, Y. 2024. Emu edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8871–8879.

Shi, Y.; Xue, C.; Liew, J. H.; Pan, J.; Yan, H.; Zhang, W.; Tan, V. Y.; and Bai, S. 2024. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8839–8849.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Tudosiu, P.-D.; Yang, Y.; Zhang, S.; Chen, F.; McDonagh, S.; Lampouras, G.; Iacobacci, I.; and Parisot, S. 2024. MULAN: A Multi Layer Annotated Dataset for Controllable Text-to-Image Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22413–22422.

Wang, X.; Darrell, T.; Rambhatla, S. S.; Girdhar, R.; and Misra, I. 2024a. Instancediffusion: Instance-level control for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6232–6242.

Wang, Z.; Li, A.; Li, Z.; and Liu, X. 2024b. Genartist: Multimodal llm as an agent for unified image generation and editing. *arXiv preprint arXiv:2407.05600*.

Wu, Z.; Kolkin, N.; Brandt, J.; Zhang, R.; and Shechtman, E. 2024. TurboEdit: Instant text-based image editing. *ECCV*.

Xiao, S.; Wang, Y.; Zhou, J.; Yuan, H.; Xing, X.; Yan, R.; Wang, S.; Huang, T.; and Liu, Z. 2024. Omnigen: Unified image generation. *arXiv preprint arXiv:2409.11340*.

Ye, H.; Zhang, J.; Liu, S.; Han, X.; and Yang, W. 2023. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*.

Zhang, K.; Mo, L.; Chen, W.; Sun, H.; and Su, Y. 2024a. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36.

Zhang, S.; Yang, X.; Feng, Y.; Qin, C.; Chen, C.-C.; Yu, N.; Chen, Z.; Wang, H.; Savarese, S.; Ermon, S.; et al. 2024b. Hive: Harnessing human feedback for instructional visual editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9026–9036.

Zhang, Z.; Xie, J.; Lu, Y.; Yang, Z.; and Yang, Y. 2025. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer. *arXiv preprint arXiv:2504.20690*.

Zhao, H.; Ma, X.; Chen, L.; Si, S.; Wu, R.; An, K.; Yu, P.; Zhang, M.; Li, Q.; and Chang, B. 2024. UltraEdit: Instruction-based Fine-Grained Image Editing at Scale. *arXiv preprint arXiv:2407.05282*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36: 46595–46623.

Zhuang, J.; Zeng, Y.; Liu, W.; Yuan, C.; and Chen, K. 2023. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. *arXiv preprint arXiv:2312.03594*.