REASONEDIT: Towards Reasoning-Enhanced Image Editing Models

Fukun Yin Shiyu Liu Yucheng Han Zhibo Wang Peng Xing Rui Wang Wei Cheng
Yingming Wang Aojie Li Zixin Yin Pengtao Chen Xianfang Zeng* Gang Yu* Daxin Jiang*
{zengxianfang, yugang, djiang}@stepfun.com
StepFun

https://github.com/stepfun-ai/Step1X-Edit

Abstract

Recent advances in image editing models have shown remarkable progress. A common architectural design couples a multimodal large language model (MLLM) encoder with a diffusion decoder, as seen in systems such as Step1X-Edit and Qwen-Image-Edit, where the MLLM encodes both the reference image and the instruction but remains frozen during training. In this work, we demonstrate that unlocking the reasoning capabilities of MLLM can further push the boundaries of editing models. Specifically, we explore two reasoning mechanisms, thinking and reflection, which enhance instruction understanding and editing accuracy. Based on that, our proposed framework enables image editing in a thinking–editing–reflection loop: the thinking mechanism leverages the world knowledge of MLLM to interpret abstract instructions, while the reflection reviews editing results, automatically corrects unintended manipulations, and identifies the stopping round. Extensive experiments demonstrate that our reasoning approach achieves significant performance gains, with improvements of ImgEdit(+4.3%), GEdit(+4.7%), and Kris(+8.2%) when initializing our DiT from the Step1X-Edit(ReasonEdit-S), and also outperforms previous open-source methods on both GEdit and Kris when integrated with Qwen-Image-Edit(ReasonEdit-Q).

1. Introduction

Image editing with diffusion models has witnessed rapid progress, moving from early mask-based approaches such as BrushNet [25] and PowerPaint [63], to instruction-driven systems like InstructPix2Pix [7] and OmniGen [56], and more recently to multimodal frameworks that integrate an MLLM encoder with a diffusion decoder, like Step1X-Edit [36] and Qwen-Image-Edit [53]. These advances have substantially improved controllability and usability,

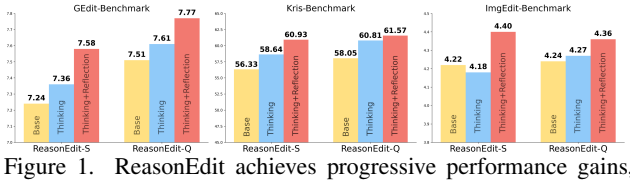

Figure 1. ReasonEdit achieves progressive performance gains, employing Thinking to interpret abstract instructions and Reflection to audit and correct the initial results.

enabling more diverse and flexible image editing. However, state-of-the-art instruction-based methods still face challenges in generalizing instructions, as most models keep MLLM encoders frozen during training. As a result, current models exhibit limited visual reasoning capabilities, which restricts their ability to handle complex or abstract instructions. More importantly, such limitations prevent them from benefiting fully from test-time scaling, a paradigm that has driven significant improvements in language models.

Turning to the visual reasoning domain, recent advances have explored reasoning-enhanced visual generation through unified understanding and generation [11, 31, 49], reflection-based refinement [29, 54], and chain-of-thought modeling [22, 51, 60]. These studies highlight the potential of reasoning for controllable and efficient generation. For instance, BAGEL [11] introduces a thinking mode that leverages the world knowledge of MLLMs to interpret abstract instructions in image generation and editing, while OmniGen2 [54] integrates reflection capabilities of MLLMs into generation. Despite these advances, most existing efforts remain centered on image generation [6, 26], leaving the application of reasoning to image editing largely underexplored. A key underlying challenge lies in the substantial hallucinations of MLLMs during paired image understanding [12, 21], particularly in capturing the differences between reference and edited results [16, 50] and in generating appropriate refined instructions for subsequent editing.

To this end, we propose ReasonEdit, a fundamental editing model with two reasoning capabilities: **thinking** and **reflection**. The former primarily transforms ambiguous, colloquial, and informal editing instructions into clear, stan-

*Corresponding author.

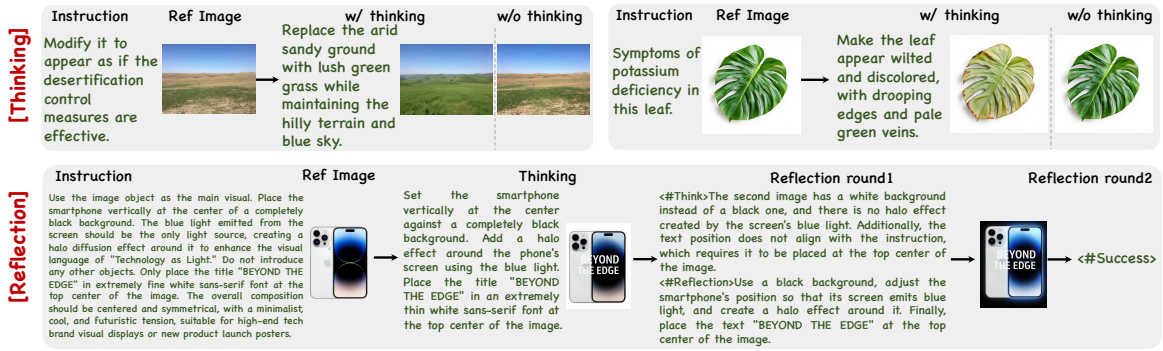

Figure 2. Illustration of the ReasonEdit's reasoning capabilities. The thinking module illustrates how a model decomposes abstract instructions into clear, actionable commands. The reflection pipeline, conversely, showcases the model's ability to perform an iterative self-correction loop, refining an intermediate generated image to achieve a more accurate final result.

dardized, and actionable directives by constructing Thinking Pairs, which are structured as abstract-to-concrete instruction pairs. The latter is designed to perform iterative self-correction, refinement, and termination during the editing process by restructuring the paired image understanding as multiple cascaded single-image understanding tasks. We achieve this by constructing Reflection Triples that form an iterative cycle defined by three core image states: <original image, editing instructions, edited image, reflection instructions, reflection-corrected image, VIEScore [28]>. To train these image editing reasoning capabilities, our architecture integrates a MLLM as the Reasoner and a DiT as the Generator. We employ a multi-stage training strategy: initially, the model is trained independently on image editing and thinking tasks, followed by a joint training phase. This progressive approach simplifies the learning objectives at each stage, leading to smoother convergence and a more effective, gradual acquisition of both editing and reasoning capabilities. Our contributions can be summarized as follows:

- A reasoning-enhanced editing model that natively supports a thinking–editing–reflection workflow. Thinking mode allows parsing original instructions leveraging the world knowledge of MLLMs, enabling the model to tackle more complex editing tasks. Reflection mode enables iterative refinement by reviewing and correcting the results of previous edits.
- A comprehensive data construction pipeline consisting of <original image, editing instructions, edited image, reflection instructions, reflection-corrected image, VIEScore>, which supports end-to-end training of the thinking–editing–reflection loop.
- A flexible training framework that demonstrates consistent performance gains by initializing our DiT from advanced models, such as ReasonEdit-S (based on Step1X-Edit) improving ImgEdit(+4.3%), GEdit(+4.7%), and Kris(+8.2%); and ReasonEdit-Q (based on Qwen-Image-Edit) yielding ImgEdit(+2.8%), GEdit(+3.4%), and Kris(+6.1%), while outperforming previous open-source methods on GEdit and Kris.

2. Related Work

2.1. Image Editing Models

Diffusion models have demonstrated remarkable progress in generative modeling, particularly for producing high-fidelity and diverse image editing results. Early approaches, such as BrushNet [25], BrushEdit [30], PowerPaint [63], and FLUX.1-Fill-dev [3], typically employ an edit-area mask together with textual instructions to achieve localized and high-quality edits. Beyond mask-based control, recent works have further explored enhancing editing controllability by incorporating multiple visual conditions. For instance, OminiControl [48], ACE [17], and ACE++ [39] unify diverse conditional signals such as depth maps and keypoints within a single model, thereby enabling more flexible and versatile editing capabilities.

While visual conditions offer precise control, they raise the usage threshold. In contrast, instruction-based models enable editing through natural language, but often struggle to align semantic understanding with fine-grained manipulation. Pioneering efforts such as Instruct-Pix2Pix [7], MagicBrush [59], UltraEdit [62], AnyEdit [58], and OmniGen [56] construct large-scale instruction–image pairs to support purely instruction-driven editing, yet still face challenges in fidelity and quality. Recent approaches address this challenge by leveraging priors from advanced text-to-image models [2, 4, 8], as in ICEdit [61], Hidream-E1 [19], and FLUX.1-Kontext-dev [5]. Another line of work integrates multimodal large language model encoders with diffusion decoders, such as Qwen2VL-Flux [37], MetaQueries [42], BLIP3-o [9], UniWorld-v1 [34], Step1X-Edit [36], and Qwen-Image-Edit [53].

Although existing models have achieved notable progress in instruction-based editing, their reliance on frozen MLLM encoders limits performance on complex or abstract instructions. Motivated by this, ReasonEdit unlocks the reasoning capability of MLLMs through joint optimization with the diffusion decoder, thereby improving semantic understanding and extending the boundaries of controllable image editing.

2.2. Reasoning-Enhanced Visual Generation

The test-time scaling paradigm has rapidly extended from language to multimodal domains, giving rise to several reasoning-enhanced visual generation models. ThinkDiff [40] introduces multimodal in-context reasoning into diffusion models via a "think-then-diffuse" inference scheme, while BAGEL [11] enables a thinking mode by jointly training visual understanding and generation tasks. In addition to pre-thinking mechanisms before generation, some works explore reflection strategies to refine outputs, such as OmniGen2 [54] and Reflect-DiT [29]. Others, including Image-CoT [60], MINT [51], and IRG [22], employ multimodal chain-of-thought reasoning to guide generation. Beyond text-to-image, GoT [14] integrates reasoning with diffusion models using large-scale reasoning-chain data for controllable generation and editing. Uni-CoT [44] further decomposes multimodal chain-of-thought learning into macro- and micro-level components with auxiliary tasks, enabling efficient training for complex reasoning. MGIE [15] fine-tunes models via MLLM guidance, providing thinking without reflection. Zero-shot frameworks like SANE [23] decompose instructions for pre-trained editors (e.g., Step1X-Edit, Qwen-Image-Edit), while CCA [18] employs an agentic pipeline with multi-image reflection. For comparison, we evaluate CCA-style baselines (denoted as CCA-Gemini/CCA-4o, using Gemini-2.5-Pro/GPT-4o paired with Step1X-v1.1 [36] and Qwen-Image-Edit [53]).

Compared with the above approaches, ReasonEdit focuses more on exploring thinking and reflection mechanisms for editing tasks, enhancing instruction understanding and editing accuracy. While Uni-CoT [44] is a concurrent work, our method adopts different base models, training data composition, and training paradigm.

3. Method

This section introduces the training data construction and the training of our model. We first elaborate on the construction of our edit reasoning data. Following this, we describe the training of the proposed REASONEDIT, presenting the model design and the multi-stage training strategy. Finally, we provide the training details.

3.1. Data Construction

To facilitate supervised fine-tuning of our reasoning model, we have developed two distinct datasets: thinking and reflection. The former consists of abstract instruction-clear multi-step decomposition instruction pairs, while the latter includes triples that encompass multiple cascaded single-image understanding tasks.

Thinking Pairs consist of abstract-to-concrete instruction pairs. Each pair links an abstract instruction, which captures a user's original request in ambiguous, colloquial, or informal language, with its corresponding set of concrete, actionable commands. The concrete counterpart translates the initial user intent into one or more precise, standardized, and executable directives. For instance, the abstract entry "symptoms of potassium deficiency in leaves" is paired with the concrete command "Render the leaves yellow and desiccate the leaf tips." For more complex requests, this structure facilitates a logical decomposition into a single, cascaded sequence of directives. As an illustration, a multifaceted request like "Make the image more dramatic with a vintage feel" is deconstructed into a single, composite instruction: "Increase the image contrast. Apply a sepia tone filter. Add a subtle vignette effect".

To construct the Thinking Pairs dataset, we devised a three-step process combining categorization, annotation, and review, leveraging advanced Vision-Language Models (VLMs) as annotators. First, we classified a large pool of raw instructions as either already clear or as abstract and complex. Then, in a two-way annotation process, we generated abstract instructions for the clear commands and decomposed the complex instructions into clear, actionable sub-directives. Finally, a rigorous review ensured that each pair met our specific abstract-to-concrete requirements. We also supplemented the dataset with a small number of simple instructions that did not require rewriting. This ensures our model can handle both complex requests by decomposing them and simple requests by outputting them directly.

The Thinking Pairs dataset is built from an initial 500k image-instruction pair pool, categorized into 112k complex and 388k simple instructions. After annotating the entire set, a rigorous review process selects 150k high-quality abstract-to-concrete pairs. Specifically, 62k of these come from simplifying complex instructions, and 88k are created by adding an abstract layer to simple ones. To ensure versatility, we include 50k pairs of simple, unedited instructions. The final dataset totals 200k carefully curated pairs.

Reflection Triples is constructed from a large collection of existing image-editing pairs, designed to facilitate a model's multi-step, cascaded reasoning capabilities. Each core triple consists of an <Input Image>, a <Generated Image>, and a <Target Image>. This structure models a chained editing process: the <Generated Image> represents an intermediate output from an initial edit on the <Input Image>, providing the crucial context for a multi-round, single-image reflection process through which the model evaluates the generated output and performs subsequent adjustments to produce the refined <Target Image>. In some instances, the generated image and the target image are identical if no further edits are required.

To mitigate hallucinations inherent in single-pass, dual-image evaluation, we designed a multi-round, single-image reflection pipeline to enable robust reflection. This process begins by generating a target image description based on the input image and instruction, which acts as a faithful and

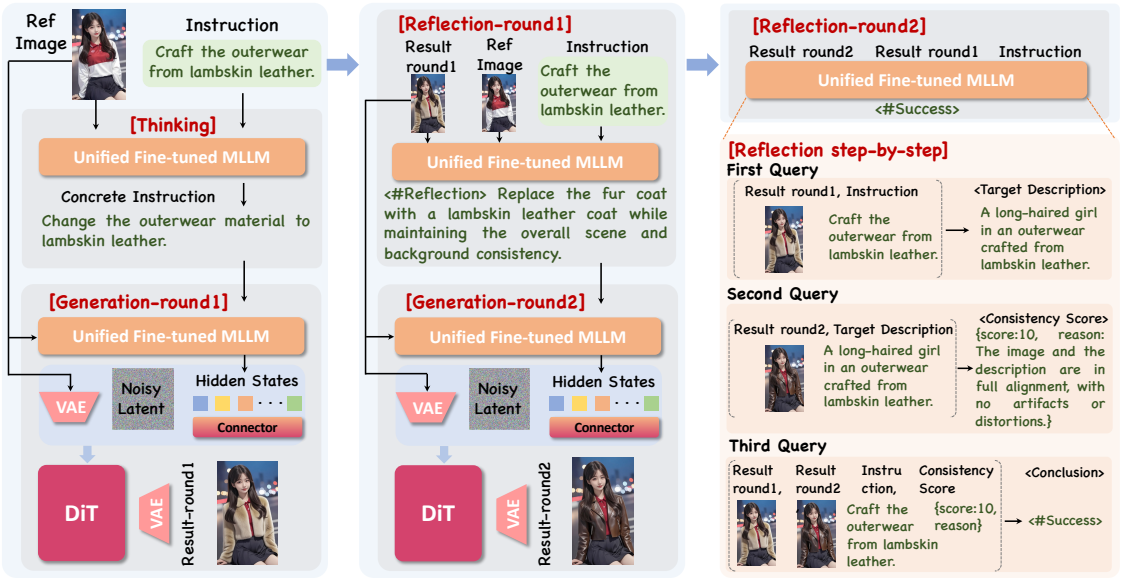

Figure 3. The model architecture and inference pipeline of REASONEDIT are structured around two core components: a Reasoner (Thinking and Reflection processes) and a DiT Generator. These modules undergo a multi-stage training process. During inference, they operate in an interleaved manner, progressively yielding more precise image editing results through their integrated reasoning capabilities.

concise blueprint for the intended outcome. A quantitative evaluation then provides a consistency score and rationale, with metrics designed to assess the presence of conflicts, omissions, and hallucinations. This comprehensive evaluation serves as a strong prior for the final reflection and decision-making phase, where the model assesses the edit's success using the original image, the generated image, and the original instruction as a basis. The process yields one of three conclusions: **Successful Edit:** Indicated by reasoning content and a <#Success> tag, signifying consistency between the generated and target images. **Refinable Edit:** For edits that are not fully successful but allow for refinement, reasoning content and a <#Reflection> tag are returned, along with a secondary editing instruction based on the generated image. **Failed Edit:** If an edit fails due to irrecoverable flaws, reasoning content and a <#Failed> tag are provided. Furthermore, the model conducts a final scoring of the generated image, assessing both semantic accuracy and image quality, to determine the optimal round at which the iterative process should terminate.

The Reflection Triples is constructed from an initial pool of 500k image-editing pairs. To diversify the modalities of intermediate images, we generate an additional 500k images using four mainstream editing methods [5, 36, 41, 46]. We then apply our previously described reflection pipeline, utilizing an advanced VLM to automate the process. After rigorous manual screening, the curation yields 180k valid data pairs, with an approximate ratio of 3:1:1 for success, reflection, and failed examples. We utilize GPT-4.1 [41] to evaluate the VIEScore for these 180k valid data pairs.

3.2. Training

With the reasoning-enhanced dataset, we utilize a multi-stage training strategy to effectively integrate reasoning and image editing capabilities into a single unified model.

3.2.1. Model Design

As shown in Fig. 3, our model integrates an MLLM as the Reasoner and a DiT [43] as a Generator. Specifically, we directly adopt Step1X-Edit [36] and Qwen-Image-Edit [53] as our base architectures, which employ Qwen2.5VL 7B Instruct [1] for text embedding and a DiT as their diffusion heads, initialized from these respective models. In contrast to the original base models, we enhance the MLLM and diffusion transformer with Thinking and Reflection capabilities on image editing. This is achieved through a multi-stage training strategy and subsequent fine-tuning on our reasoning-enhanced dataset, thereby progressively refining the model's performance. It is important to note that, while Step1X-Edit and Qwen-Image-Edit serve as our chosen implementations, our proposed method is broadly applicable across various image editing approaches.

3.2.2. Multi-stage Training

Prior work highlights that such reconciliation often necessitates dedicated architectural advancements, for instance, in vision encoders [38, 45, 52], to mitigate conflicts during early joint training on both understanding and generation. To address these complex dynamics and effectively integrate enhanced reasoning with generative processes, we adopt a multi-stage training strategy. This progressive approach decomposes the intricate joint optimization into simpler, focused tasks: initially cultivating the MLLM's ex-

plicit Thinking and Reflection, subsequently adapting the Generator (DiT) to these refined MLLM on image editing, and culminating in a comprehensive joint fine-tuning of both components to achieve superior overall performance.

Reasoning Learning Stage. This initial stage is dedicated to cultivating the MLLM's explicit Thinking and Reflection capabilities tailored for image editing tasks. To efficiently adapt the model while mitigating catastrophic forgetting of its foundational knowledge, and to isolate reasoning training, we employ Low-Rank Adaptation (LoRA) [20] on the linear layers in attention modules. During this phase, the DiT remains frozen. Training is conducted on the constructed Thinking Pairs and Reflection Triples datasets ($cf.$ Sec. 3.1), optimizing with a standard Next Token Prediction (NTP) loss,

$$\mathcal{L}_{\text{NTP}} = \mathbb{E}_{t_i}\left[-\sum_{k=1}^{L}\log p_\theta(t_k|t_1, t_2, \cdots, t_{k-1})\right] \quad (1)$$

where t_k represents the k-th token in a sequence of length L, and p_θ is the probability predicted by the MLLM parameterized by θ.

Edit Learning Stage. Following the dedicated tuning of the MLLM's reasoning abilities, this stage focuses on adapting the Generator, specifically the DiT model. To leverage the MLLM's refined contextual understanding without interference, its parameters are kept frozen throughout this phase. The DiT is trained using a flow matching loss [35], with a dual objective that encompasses both text-to-image (T2I) generation and direct image editing tasks. Including T2I data is crucial; their significantly larger scale and broader domain coverage are instrumental in enriching the model's general generative knowledge, which in turn substantially improves its proficiency in diverse editing scenarios. The flow matching loss is formulated as,

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t\sim U(0,1),x_0\sim\mathcal{D},x_1\sim\mathcal{N}(0,I),c}\|u_t(x|c) - v_t(x|x_0,c)\|_2^2 \quad (2)$$

where t is uniformly sampled from $[0, 1]$, x_0 is a data point from the dataset $\mathcal{D}$, x_1 is standard Gaussian noise, and c represents the conditioning information (e.g., a text or a reference image). The DiT model u_t is trained to predict the target vector field $x_1 - x_0$ at the interpolated point $x_t = (1-t)x_0 + tx_1$.

Unified Tuning Stage. After the preceding stages, this final stage unifies and jointly fine-tunes both the MLLM and the DiT. This comprehensive joint optimization is crucial for ensuring that the understanding and generative processes seamlessly complement each other. The joint training loss for this stage is formulated as,

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{FM}} + \omega_{\text{NTP}} \cdot \mathcal{L}_{\text{NTP}} \quad (3)$$

3.2.3. Training Details

We utilized Step1X-Edit V1.1 [47] and Qwen-Image-Edit [53] as our pretrained models. The first reasoning learning stage involved training the MLLM on 32 H800 GPUs (4 nodes, 8 GPUs/node) for 16 hours, completing 50,000 steps with an initial learning rate of 1×10^{-4}. The second edit learning stage scaled to 128 GPUs (16 nodes, 8 GPUs/node), training for 38.9 hours and 28,000 steps at a learning rate of 1×10^{-5}. During this stage, we use in-house 14.4M T2I samples and 2.4M image editing samples for training. The final stage consisted of 20 hours of training, completing 12,000 steps with a learning rate of 6×10^{-6} and the NTP loss weight ω_{NTP} of 0.1. Similar to BAGEL [11] and Mogao [33], during this stage, FlexAttention [13] and a packed data format [10] were utilized to support efficient hybrid training for both understanding and generation tasks, especially on the Reflection Triples. To optimize training performance and scalability, distributed training employed several parallelization strategies. Specifically, the MLLM and the Connector utilized *sequence parallelism* and *DeepSpeed Ulysses* [24]. For the DiT, both *tensor parallelism* and *sequence parallelism* were applied, enabling effective scaling across multiple nodes and GPUs and accelerating the training process.

4. Experiments

4.1. Experimental Settings

Benckmark. We conduct our experiments on three widely-used benchmarks: GEdit-Bench [36] and ImgEdit-Bench [57] for evaluating broad and comprehensive foundational image editing capabilities, and KRIS-Bench [55] for assessing a model's advanced reasoning skills and ability to interpret abstract instructions. These benchmarks collectively enable a thorough evaluation of our model's performance, ranging from foundational editing tasks to complex, abstract reasoning challenges.

Metrics. For the GEdit-Bench, we evaluate performance using three metrics-Semantic Consistency (SQ), Perceptual Quality (PQ), and an Overall Score (O)-which are automatically assessed by VIEScore [27] using GPT-4.1. On the ImgEdit-Bench, we use GPT-4.1 to assign 1-5 ratings across three dimensions-instruction adherence, image-editing quality, and detail preservation-where the final score for the latter two is capped by instruction adherence. On the KRIS-Bench, we use GPT-4o to assign 1-5 ratings across four dimensions-Visual Consistency, Visual Quality, Instruction Following, and the novel Knowledge Plausibility, which assesses consistency with real-world knowledge.

4.2. Experimental Results

Quantitative results are first reported on GEdit-Bench [36]and ImgEdit-Bench [57] to assess foundational editing capabilities ($cf.$ Sec. 4.2.1), with the evaluation then shifting to the more complex KRIS-Bench [55] for an assessment of abstract reasoning skills ($cf.$ Sec. 4.2.2).

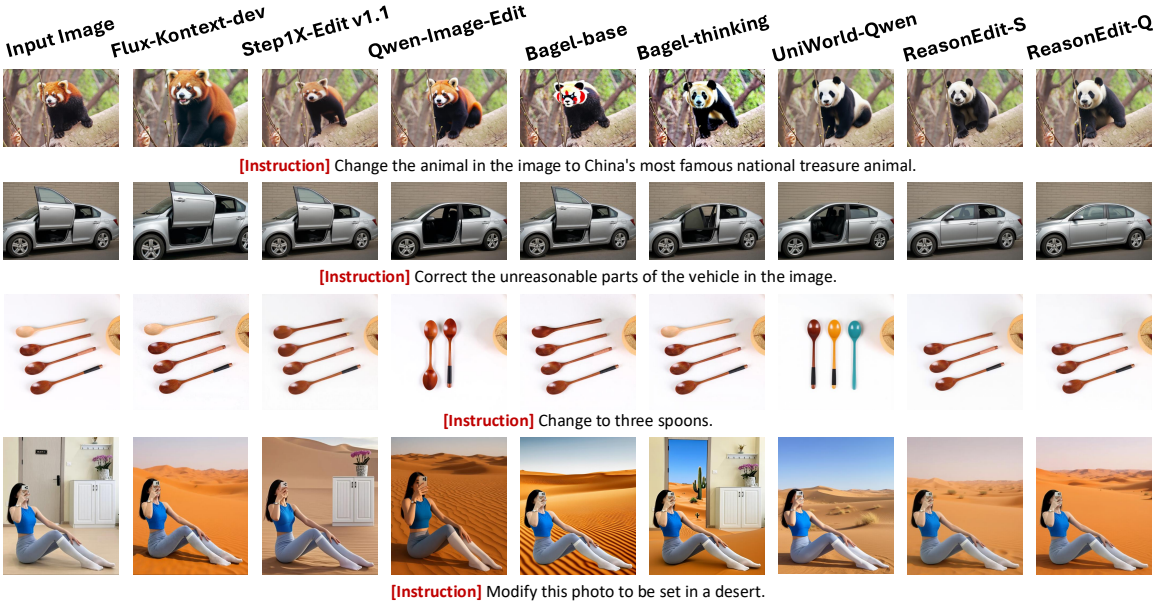

Figure 4. Comprehensive qualitative evaluation of leading image editing models. The results demonstrate that our proposed approach, which incorporates thinking and reflection mechanisms, significantly outperforms the editing model.

4.2.1. Evaluation on GEdit-Bench and ImgEdit-Bench

As shown in Table 1, our method achieves superior performance on the foundational instruction benchmarks ImgEdit-Bench [57] and GEdit-Bench [36]. Both ReasonEdit-S and ReasonEdit-Q exhibit consistent and substantial improvements over their respective base models, achieving gains of +4.3% and +4.7% on ImgEdit and GEdit for ReasonEdit-S, and +2.8% and +3.4% for ReasonEdit-Q, respectively. Importantly, although the two methods are built upon different underlying models - Step1X-Edit v1.1 for ReasonEdit-S and Qwen-Image-Edit for ReasonEdit-Q - both variants significantly surpass the performance of their corresponding underlying editors. ReasonEdit-S ranks third on GEdit-Bench, outperforming Qwen-Image-Edit, while ReasonEdit-Q achieves the highest overall score. On ImgEdit-Bench, ReasonEdit-S and ReasonEdit-Q place second and third among all open-source models, trailing the top entry by only 0.08 and 0.12 points, respectively.

GEdit-Bench and ImgEdit-Bench primarily evaluate a model's foundational editing capabilities. While our thinking and reflection mechanisms provide performance gains, their full impact may be less pronounced on these relatively simple tasks compared to more complex ones. This is consistent with the design of our dataset, where the thinking and reflection modules are specifically tailored for complex instructions and multi-step editing.

As shown in Fig. 4, a qualitative comparison demonstrates that our approach excels at precisely altering target areas while faithfully maintaining the integrity of unedited regions, such as backgrounds, facial features, and hairstyles. This capability addresses a key challenge in image editing by mitigating common failures related to consistency and fidelity, resulting in stable performance and accurate responsiveness to a wide range of commands.

4.2.2. Evaluation on Kris-Bench

On the KRIS-Bench [55], the proposed approach demonstrates strong performance among open-source models, including those that also employ a thinking-based mechanism (e.g., BAGEL-Thinking), and surpasses several closed-source methods. This superiority is further substantiated by substantial improvements over their respective base models (w/o thinking, w/o reflection): ReasonEdit-S (built upon Step1X-Edit v1.1 [36]) attains a performance gain of +8.2%, while ReasonEdit-Q (derived from Qwen-Image-Edit [53]) achieves an improvement of +6.1%. Furthermore, ReasonEdit exhibits broader generalizability compared to existing methods. While Uni-CoT employs sequential execution with background knowledge that aids complex evaluations like KRIS-Bench, it shows limited improvements on standard benchmarks (e.g., GEdit-Bench and ImgEdit-Bench). In contrast, ReasonEdit avoids these limitations, delivering consistent performance gains across both complex and standard editing tasks.

The performance gains are attributed to the method's ability to simplify abstract and difficult editing tasks into clear, actionable steps for the editing model. Furthermore, the reflection pipeline provides a crucial mechanism to analyze the correctness of an edit and formulate strategies for improvement. This iterative process of self-correction allows the model to identify and rectify subtle errors, effectively mitigating hallucination and improving overall fidelity. The method's demonstrated effectiveness on both

Table 1. Comprehensive quantitative evaluation of leading image editing models. Our approach achieves significant performance gains over its base models and achieves state-of-the-art performance among open-source models on both GEdit and Kris (with ReasonEdit-Q), while also proving to be highly competitive with several closed-source models.

	Models	GEdit-Bench			KRIS-Bench				ImgEdit-Bench
		Semantic Consistency	Quality	**Overall**	Factual Knowledge	Conceptual Knowledge	Procedural Knowledge	**Overall**	**Overall**
close-source models	Gemini 2 flash (Apr. 2025)	6.87	7.44	6.51	65.26	59.65	62.90	62.41	-
	Gemini 2.5 flash (Sep. 2025)	8.25	8.29	7.89	77.03	78.29	75.93	77.29	4.30
	Doubao (seed edit 1.6, Apr. 2025)	7.22	7.89	6.98	63.30	62.23	54.17	60.70	-
	Doubao (Seedream 4.0, Aug. 2025)	9.17	7.95	8.40	78.10	76.86	76.93	77.31	4.46
	GPT4o (Apr. 2025)	7.74	8.13	7.49	79.80	81.37	78.32	80.09	-
	GPT4o (Sep. 2025)	8.74	7.67	8.01	81.16	78.24	77.09	79.00	4.30
open-source models	ICEdit [61]	4.94	7.39	4.87	46.99	42.73	27.76	40.70	3.05
	Omnigen [56]	5.88	5.87	5.01	33.11	28.02	23.89	28.85	2.96
	Omnigen 2 [54]	7.16	6.77	6.41	57.36	44.20	47.79	49.71	3.44
	BAGEL-thinking [11]	7.70	6.51	6.66	66.18	61.92	49.02	60.18	3.56
	BAGEL [11]	7.48	6.80	6.60	60.26	55.86	51.69	56.21	3.20
	Uniworld-V1 [34]	4.93	7.43	4.85	47.71	44.80	47.92	50.27	3.26
	Hidream-I1 (E1) [19]	5.66	6.06	5.01	43.31	50.05	37.64	44.72	3.17
	Hidream-E1.1 [19]	7.15	6.65	6.42	43.52	44.71	36.08	42.25	3.97
	Flux-Kontext-dev [5]	7.16	7.37	6.51	53.28	50.36	42.53	49.54	3.97
	Uni-CoT [44]	7.91	6.24	6.74	71.85	67.16	63.68	**68.00**	3.65
	CCA-GPT-4o [18]	7.90	7.41	7.43	58.73	67.19	47.89	59.62	4.27
	CCA-Gemini-2.5-Pro [18]	7.99	7.42	7.54	65.73	67.51	53.36	63.53	4.33
	UniWorld-FLUX.1-Kontext-Dev [32]	7.28	7.49	6.74	55.50	51.39	43.76	51.04	4.02
	UniWorld-Qwen-Image-Edit [32]	8.36	7.87	7.76	61.72	56.38	46.69	55.98	**4.48**
	Step1X-Edit v1.1 [36]	7.66	7.35	6.97	53.05	54.34	44.66	51.59	3.90
	ReasonEdit-S (base)	7.77	7.65	7.24	58.23	60.55	46.21	56.33	4.22
	ReasonEdit-S (thinking)	8.02	7.64	7.36 (+1.7%)	59.79	62.76	49.78	58.64 (+4.1%)	4.18 (-0.9%)
	ReasonEdit-S (thinking+reflection)	8.18	7.85	7.58 (+4.7%)	62.44	65.72	50.42	60.93 (+8.2%)	4.40 (+4.3%)
	Qwen-Image-Edit [53]	8.00	7.86	7.56	61.47	56.79	47.07	56.15	4.27
	ReasonEdit-Q (base)	8.12	7.94	7.51	62.29	62.22	44.53	58.05	4.24
	ReasonEdit-Q (thinking)	8.20	7.96	7.61 (+1.3%)	62.44	64.49	52.02	60.81 (+4.8%)	4.27 (+0.7%)
	ReasonEdit-Q (thinking+reflection)	8.34	7.97	**7.77** (+3.4%)	63.92	64.85	52.41	61.57 (+6.1%)	4.36 (+2.8%)

complex and simple tasks (cf. Sec. 4.2.1) proves its versatility and robust generalization.

As shown in Fig. 4, many methods often misinterpret or fail to respond correctly to abstract or complex instructions. Our proposed thinking module effectively aids the editing model in understanding such instructions and executing them accurately. Furthermore, the reflection pipeline enhances this process by enabling the model to identify and rectify subtle errors, formulate precise refinement strategies, and prevent the compounding of mistakes that are common in multi-step editing tasks. Simultaneously, many models struggle with maintaining consistency in complex scenarios, often leading to unintended alterations in unedited regions because they lack a robust understanding of the entire scene's structure. In contrast, our approach ensures high consistency by faithfully preserving elements that should remain unchanged.

4.3. Ablation Studies

To systematically evaluate the contribution of each component of the proposed method, a series of ablation studies on ReasonEdit-S (built upon Step1X-Edit v1.1 [36] and the MLLM Qwen2.5VL 7B Instruct, hereafter Qwen) using KRIS-Bench, as its abstract and challenging nature makes it an ideal testbed for verifying the reasoning and reflection capabilities of the model.

Impact of Multi-Stage Training. To evaluate the contribution of the reasoning learning stage, we compare the performance of the Pre-trained Generator(Step1X-Edit v1.1 [36]) when integrated with either a base (untuned) Qwen model or a fine-tuned Qwen model. When the Pre-trained Generator is augmented with the base Qwen model leveraging our thinking and reflection mechanism, only a marginal performance gain of 0.82 points is observed. In contrast, fine-tuning Qwen on our reasoning data consistently and significantly outperforms this base configuration. This highlights that foundational multimodal large language models, without domain-specific adaptation, struggle to effectively grasp the nuances of image editing, thereby underscoring the critical necessity of tailoring the MLLM to these specific demands. After the edit learning stage, in isolation, the Base Generator achieves a degree of performance

Table 2. Ablation of Multi-Stage Training. This table evaluates the performance contributions of each stage in the training pipeline, from the pre-trained baseline to the final unified model, highlighting the cumulative benefits of fine-tuning the generator and reasoning modules at each step.

Methods	KRIS-Bench			
	Factual Knowledge	Conceptual Knowledge	Procedural Knowledge	**Overall**
Pre-trained Generator (Step1X-Edit V1.1 [36])	53.05	54.34	44.66	51.59
Pre-trained Generator + Qwen Reasoning	54.05	57.44	41.26	52.41
Pre-trained Generator + Qwen-tuned Reasoning	56.32	62.00	46.17	56.24
Base Generator W/O Reasoning	55.80	55.28	43.78	52.74
Base Generator + Qwen-tuned Reasoning	60.54	62.16	48.26	58.29
Unified Tuned (ReasonEdit-S)	62.44	65.72	50.42	**60.93**

improvement over the Pre-trained Generator, demonstrating its role in adapting the generative capabilities. Finally, this multi-stage strategy culminates in the optimal performance of the unified training, providing a substantial performance increase from the Base Generator + Qwen-tuned Reasoning model to the Unified Tuned model (58.29 vs. 60.93), validating the synergistic benefits of training the entire pipeline as a whole.

Table 3. Ablation Study on the Contributions of the Thinking and Reflection Modules. The table shows the performance of four model variants on the KRIS-Bench, demonstrating the benefits of each component and the synergy of their combination.

Methods	KRIS-Bench			
	Factual Knowledge	Conceptual Knowledge	Procedural Knowledge	Overall
ReasonEdit-S (w/o thinking, w/o reflection)	58.23	60.55	46.21	56.33
ReasonEdit-S (w/ thinking, w/o reflection)	59.79	62.76	49.78	58.64
ReasonEdit-S (w/o thinking, w/ reflection)	61.40	64.16	48.16	59.39
ReasonEdit-S (w/ thinking, w/ reflection)	62.44	65.72	50.42	**60.93**

Ablation of Thinking and Reflection. To understand the individual and combined contributions of the thinking and reflection modules, four variants are compared: (1) a baseline model without either module; (2) a model with only the thinking module; (3) a model with only the reflection module; and (4) the full model incorporating both. The results on KRIS-Bench (see Tab. 3) show a gradual improvement in performance with the addition of each component. The thinking module alone provides a significant performance boost, confirming its effectiveness in handling complex instructions. The thinking + reflection module proves beneficial across all benchmarks (see Tab. 1), as it effectively rectifies errors. The full model, with both modules integrated, achieves the highest scores, highlighting the synergistic relationship between understanding an instruction and correcting subsequent errors.

Table 4. Ablation Study on Reflection Pipelines. The table compares three different reflection mechanisms on KRIS-Bench, highlighting the effectiveness of the proposed multi-round pipeline.

Methods	KRIS-Bench			
	Factual Knowledge	Conceptual Knowledge	Procedural Knowledge	Overall
Base Generator	55.80	55.28	43.78	52.74
Base Generator + dual-image pipeline	52.97	61.84	41.12	53.79
Base Generator + single-image pipeline	54.81	56.92	43.70	53.04
Base Generator + our multi-round pipeline	60.54	62.16	48.26	**58.29**

Comparison of Reflection Pipelines. To ensure consistency in the DiT parameters, this ablation study is conducted by combining the Base Generator (the DiT after the edit learning stage) with each reflection pipeline. Tab. 4 compares three distinct approaches to the reflection process—a dual-image pipeline, a pure single-image pipeline, and the proposed multi-round prior pipeline. The dual-image pipeline, which relies on a direct comparison between the initial input and the generated output, is often prone to hallucinations. Conversely, a pure single-image approach struggles with tasks that require a clear before-and-after comparison, such as Portrait Beautification or motion/expression-related edits. As shown in the table, the proposed multi-round single-image prior pipeline is superior. This is attributed to the method's ability to combine the benefits of both approaches, allowing it to perform a self-correction loop on the generated image itself while leveraging key prior information from the multi-round process.

Reflection Performance Curve. The effect of varying reflection rounds is evaluated on KRIS-Bench using ReasonEdit-S (see Tab. 5). The results show that incorporating reflection consistently improves performance over the Thinking-only baseline (58.64). Specifically, two reflection rounds yield a score of 60.93, while extending to three or four rounds brings only marginal improvements (+0.06 and +0.14, respectively) with higher computational cost. We further evaluate a naive re-roll baseline (see Tab. 6). This simpler strategy yields only minor gains, peaking at 59.24 after three attempts, and even drops to 59.09 at four attempts with increased cost. The comparison indicates that the targeted reflection mechanism markedly outperforms an unguided re-roll strategy, highlighting the role of structured reasoning in iterative error correction (see Fig. 5).

Table 5. Performance-efficiency curve of reflection rounds.

Reflection Rounds	0 (Thinking)	1	2	3	4
Performance	58.64	60.08	60.93	60.99	61.07
Time(s)	40	80	120	160	200

Table 6. Performance-efficiency curve of re-roll times.

Re-roll Times	0 (Thinking)	1	2	3	4
Performance	58.64	58.84	59.00	59.24	59.09
Time(s)	39	78	117	156	195

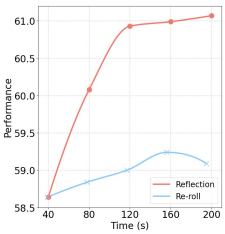

Figure 5. Performance-Efficiency Curve.

5. Conclusion

In this work, we present ReasonEdit, a fundamental image editing framework that demonstrates the crucial role of explicit reasoning in achieving robust and versatile performance. The proposed method introduces a novel pipeline with two core capabilities: thinking and reflection. By training these capabilities on a curated collection of Thinking Pairs and Reflection Triples, the framework learns to convert abstract user requests into actionable commands and to perform self-correction in an iterative loop. Extensive experiments on a range of benchmarks validate the efficacy of this approach, with the model achieving state-of-the-art performance among open-source methods while remaining highly competitive with several closed-source models. This work provides a new perspective on reasoning-enhanced image editing, showing that a structured pipeline for instruction understanding and self-correction is vital for building models that can handle both simple and complex editing tasks with high fidelity and consistency.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 4

[2] Black Forest Labs. Flux.1 [dev]. https://huggingface.co/black-forest-labs/FLUX.1-dev, 2024. 2

[3] Black Forest Labs. Flux.1 fill [dev]. https://huggingface.co/black-forest-labs/FLUX.1-Fill-dev, 2024. Accessed: 2025-04-19. 2

[4] Black Forest Labs. Flux.1 [schnell]. https://huggingface.co/black-forest-labs/FLUX.1-schnell, 2024. 2

[5] BlackForestLabs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. 2, 4, 7

[6] Manuel Brack, Felix Friedrich, Katharia Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinário Passos. Ledits++: Limitless image editing using text-to-image models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8861–8870, 2024. 1

[7] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18392–18402, 2022. 1, 2

[8] Qi Cai, Jingwen Chen, Yang Chen, Yehao Li, Fuchen Long, Yingwei Pan, Zhaofan Qiu, Yiheng Zhang, Fengbin Gao, Peihan Xu, et al. Hidream-i1: A high-efficient image generative foundation model with sparse diffusion transformer. *arXiv preprint arXiv:2505.22705*, 2025. 2

[9] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025. 2

[10] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, et al. Patch n'pack: Navit, a vision transformer for any aspect ratio and resolution. *Advances in Neural Information Processing Systems*, 36:2252–2274, 2023. 5

[11] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025. 1, 3, 5, 7

[12] Peng Ding, Jingyu Wu, Jun Kuang, Dan Ma, Xuezhi Cao, Xunliang Cai, Shi Chen, Jiajun Chen, and Shujian Huang. Hallu-pi: Evaluating hallucination in multi-modal large language models within perturbed inputs. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 10707–10715, 2024. 1

[13] Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flex attention: A programming model for generating optimized attention kernels. *arXiv preprint arXiv:2412.05496*, 2024. 5

[14] Rongyao Fang, Chengqi Duan, Kun Wang, Linjiang Huang, Hao Li, Shilin Yan, Hao Tian, Xingyu Zeng, Rui Zhao, Jifeng Dai, Xihui Liu, and Hongsheng Li. Got: Unleashing reasoning capability of multimodal large language model for visual generation and editing. *arXiv preprint arXiv:2503.10639*, 2025. 3

[15] 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. 3

[16] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer, 2024. 1

[17] Zhen Han, Zeyinzi Jiang, Yulin Pan, Jingfeng Zhang, Chaojie Mao, Chen-Wei Xie, Yu Liu, and Jingren Zhou. ACE: All-round creator and editor following instructions via diffusion transformer. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[18] Tiankai Hang, Shuyang Gu, Dong Chen, Xin Geng, and Baining Guo. Cca: collaborative competitive agents for image editing. *Frontiers of Computer Science*, 19(11):1911367, 2025. 3, 7

[19] HiDream-ai. Hidream-e1. https://github.com/HiDream-ai/HiDream-E1, 2025. 2, 7

[20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 5

[21] Wen Huang, Hongbin Liu, Minxin Guo, and Neil Gong. Visual hallucinations of multi-modal large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9614–9631, 2024. 1

[22] Wenxuan Huang, Shuang Chen, Zheyong Xie, Shaosheng Cao, Shixiang Tang, Yufan Shen, Qingyu Yin, Wenbo Hu, Xiaoman Wang, Yuntian Tang, et al. Interleaving reasoning for better text-to-image generation. *arXiv preprint arXiv:2509.06945*, 2025. 1, 3

[23] Ekaterina Iakovleva, Fabio Pizzati, Philip Torr, and Stéphane Lathuilière. Specify and edit: Overcoming ambiguity in text-based image editing. *arXiv preprint arXiv:2407.20232*, 2024. 3

[24] Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Reza Yazdani Aminadabi, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. System optimiza-

tions for enabling training of extreme long sequence transformer models. In *Proceedings of the 43rd ACM Symposium on Principles of Distributed Computing*, page 121–130, New York, NY, USA, 2024. Association for Computing Machinery. 5

[25] Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *European Conference on Computer Vision*, pages 150–168. Springer, 2024. 1, 2

[26] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6007–6017, 2023. 1

[27] Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation. *arXiv preprint arXiv:2312.14867*, 2023. 5

[28] Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12268–12290, 2024. 2

[29] Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Arsh Koneru, Yusuke Kato, Kazuki Kozuka, and Aditya Grover. Reflect-dit: Inference-time scaling for text-to-image diffusion transformers via in-context reflection. *arXiv preprint arXiv:2503.12271*, 2025. 1, 3

[30] Yaowei Li, Yuxuan Bian, Xu Ju, Zhaoyang Zhang, Ying Shan, and Qiang Xu. Brushedit: All-in-one image inpainting and editing. *ArXiv*, abs/2412.10316, 2024. 2

[31] Zijie Li, Henry Li, Yichun Shi, Amir Barati Farimani, Yuval Kluger, Linjie Yang, and Peng Wang. Dual diffusion for unified image generation and understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2779–2790, 2025. 1

[32] Zongjian Li, Zheyuan Liu, Qihui Zhang, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Yang Ye, Wangbo Yu, Yuwei Niu, and Li Yuan. Uniworld-v2: Reinforce image editing with diffusion negative-aware finetuning and mllm implicit feedback. *arXiv preprint arXiv:2510.16888*, 2025. 7

[33] Chao Liao, Liyang Liu, Xun Wang, Zhengxiong Luo, Xinyu Zhang, Wenliang Zhao, Jie Wu, Liang Li, Zhi Tian, and Weilin Huang. Mogao: An omni foundation model for interleaved multi-modal generation. *arXiv preprint arXiv:2505.05472*, 2025. 5

[34] Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, et al. Uniworld: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*, 2025. 2, 7

[35] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. 5

[36] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, Guopeng Li, Yuang Peng, Quan Sun, Jingwei Wu, Yan Cai, Zheng Ge, Ranchen Ming, Lei Xia, Xianfang Zeng, Yibo Zhu, Binxing Jiao, Xiangyu Zhang, Gang Yu, and Daxin Jiang. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025. 1, 2, 3, 4, 5, 6, 7

[37] Pengqi Lu. Qwen2vl-flux: Unifying image and text guidance for controllable image generation, 2024. 2

[38] Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai Yu, et al. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7739–7751, 2025. 4

[39] Chaojie Mao, Jingfeng Zhang, Yulin Pan, Zeyinzi Jiang, Zhen Han, Yu Liu, and Jingren Zhou. Ace++: Instruction-based image creation and editing via context-aware content filling. *arXiv preprint arXiv:2501.02487*, 2025. 2

[40] Zhenxing Mi, Kuan-Chieh Wang, Guocheng Qian, Hanrong Ye, Runtao Liu, Sergey Tulyakov, Kfir Aberman, and Dan Xu. I think, therefore i diffuse: Enabling multimodal in-context reasoning in diffusion models. In *Forty-second International Conference on Machine Learning*, 2025. 3

[41] OpenAI. Introducing 4o image generation, 2025. 4

[42] Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, et al. Transfer between modalities with metaqueries. *arXiv preprint arXiv:2504.06256*, 2025. 2

[43] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 4

[44] Luozheng Qin, Jia Gong, Yuqing Sun, Tianjiao Li, Mengping Yang, Xiaomeng Yang, Chao Qu, Zhiyu Tan, and Hao Li. Uni-cot: Towards unified chain-of-thought reasoning across text and vision. *arXiv preprint arXiv:2508.05606*, 2025. 3, 7

[45] Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2545–2555, 2025. 4

[46] Yichun Shi, Peng Wang, and Weilin Huang. Seededit: Align image re-generation to image editing. *arXiv preprint arXiv:2411.06686*, 2024. 4

[47] StepFun AI. Step1x-edit v1.1 diffusers. https://huggingface.co/stepfun-ai/Step1X-Edit-v1p1-diffusers, 2025. Accessed: 2025-09-23. 5

[48] Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. Ominicontrol: Minimal and universal control for diffusion transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. 2

[49] NextStep Team, Chunrui Han, Guopeng Li, Jingwei Wu, Quan Sun, Yan Cai, Yuang Peng, Zheng Ge, Deyu Zhou,

Haomiao Tang, et al. Nextstep-1: Toward autoregressive image generation with continuous tokens at scale. *arXiv preprint arXiv:2508.10711*, 2025. 1

[50] Fei Wang, Xingyu Fu, James Y. Huang, Zekun Li, Qin Liu, Xiaogeng Liu, Mingyu Derek Ma, Nan Xu, Wenxuan Zhou, Kai Zhang, Tianyi Lorena Yan, Wenjie Jacky Mo, Hsiang-Hui Liu, Pan Lu, Chunyuan Li, Chaowei Xiao, Kai-Wei Chang, Dan Roth, Sheng Zhang, Hoifung Poon, and Muhao Chen. Muirbench: A comprehensive benchmark for robust multi-image understanding. In *The Thirteenth International Conference on Learning Representations*, 2025. 1

[51] Yi Wang, Mushui Liu, Wanggui He, Longxiang Zhang, Ziwei Huang, Guanghao Zhang, Fangxun Shu, Zhong Tao, Dong She, Zhelun Yu, et al. Mint: Multi-modal chain of thought in unified generative models for enhanced image generation. *arXiv preprint arXiv:2503.01298*, 2025. 1, 3

[52] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12966–12977, 2025. 4

[53] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. 1, 2, 3, 4, 5, 6, 7

[54] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025. 1, 3, 7

[55] Yongliang Wu, Zonghui Li, Xinting Hu, Xinyu Ye, Xianfang Zeng, Gang Yu, Wenbo Zhu, Bernt Schiele, Ming-Hsuan Yang, and Xu Yang. Kris-bench: Benchmarking next-level intelligent image editing models. *arXiv preprint arXiv:2505.16707*, 2025. 5, 6

[56] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13294–13304, 2025. 1, 2, 7

[57] Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275*, 2025. 5, 6

[58] Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26125–26135, 2025. 2

[59] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023. 2

[60] Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Ziyu Guo, Haoquan Zhang, Manyuan Zhang, Jiaming Liu, Peng Gao, and Hongsheng Li. Let's verify and reinforce image generation step by step. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 28662–28672, 2025. 1, 3

[61] Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer. *arXiv*, 2025. 2, 7

[62] Haozhe Zhao, Xiaojian Shawn 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. *Advances in Neural Information Processing Systems*, 37:3058–3093, 2024. 2

[63] Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. In *European Conference on Computer Vision*, pages 195–211. Springer, 2024. 1, 2