

DreamLite: A Lightweight On-Device Unified Model for Image Generation and Editing

Kailai Feng, Yuxiang Wei^(✉), Bo Chen, Yang Pan,
Songwei Liu, Chenqian Yan, Yuan Gao^(✉), and Wangmeng Zuo

ByteDance, Shanghai, China
Project Page: <https://carlofkl.github.io/dreamlite/>

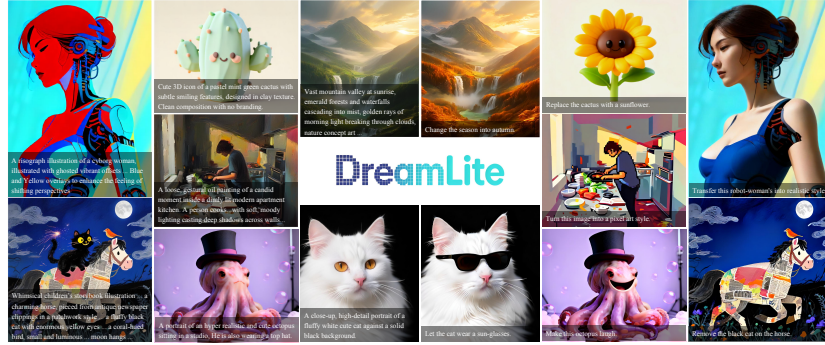


Fig. 1: The proposed DreamLite enables efficient visual creation on mobile devices, seamlessly supporting both image generation (left) and editing (right) within a unified architecture.

Abstract. Diffusion models have made significant progress in both text-to-image (T2I) generation and text-guided image editing. However, these models are typically built with billions of parameters, leading to high latency and increased deployment challenges. While on-device diffusion models improve efficiency, they largely focus on T2I generation and lack support for image editing. In this paper, we propose DreamLite, a **compact unified on-device diffusion model (0.39B)** that supports both T2I generation and text-guided image editing within a single network. DreamLite is built on a pruned mobile U-Net backbone and unifies conditioning through in-context spatial concatenation in the latent space. It concatenates images horizontally as input, using a (target | blank) configuration for generation tasks and (target | source) for editing tasks. To stabilize the training of this compact model, we introduce a task-progressive joint pretraining strategy that sequentially targets T2I, editing, and joint tasks. After high-quality SFT and reinforcement learning, DreamLite achieves GenEval (0.72) for image generation and ImgEdit (4.11) for image editing, outperforming existing on-device models and remaining competitive with several server-side models. By employing step

distillation, we further reduce denoising processing to just 4 steps, enabling our DreamLite could generate or edit a 1024×1024 image in less than 3s on iPhone 17 pro. To the best of our knowledge, DreamLite is the first unified on-device diffusion model that supports both image generation and image editing. Our code and model are available at [here](#).

Keywords: On-device GenAI, Unified Diffusion Model, RLHF, DMD

1 Introduction

Recent advancements in large-scale diffusion models, such as FLUX [21, 23], HunyuanImage 3.0 [6], Qwen Image [43] and Seedream [11, 14, 37] have achieved remarkable progress in both text-to-image (T2I) generation and text-guided editing (I2I). Despite their superior semantic alignment and visual fidelity, these models typically rely on massive backbones with billions of parameters and iterative denoising processes. For instance, FLUX. [23] scales its DiT backbone to 12B parameters, imposing prohibitive memory requirements and high inference latency that preclude efficient deployment on consumer-grade devices. To enhance efficiency, recent research has explored lightweight architectures such as SANA [47], DeepGen1.0 [42] and VIBE [1], which typically utilize backbones on the order of ~ 2 B parameters. However, achieving stable, real-time performance with these models on mobile hardware remains a significant challenge.

To improve accessibility, several works [16, 17, 24, 53] focus on deploying compact diffusion models directly on mobile devices. For example, SnapFusion [24], Mobile Diffusion [53] and SnapGen [16] leverage lightweight U-Net backbones to achieve quality–efficiency trade-offs for on-device T2I generation. More recently, SnapGen++ [17] has explored the potential of Diffusion Transformer for efficient mobile generation. However, these approaches predominantly focus on T2I generation and lack support for image editing. In practice, creators demand a unified experience that seamlessly integrates “generate” and “edit” functionalities within a single application. Furthermore, deploying two separate models significantly increases system complexity and resource consumption, particularly in memory-constrained devices.

In this paper, we introduce DreamLite, a unified and compact diffusion model capable of performing both image generation and editing within a single network. Following SnapGen [16], we adopt a pruned UNet backbone and extend it for multi-task learning via an in-context conditioning mechanism. Specifically, we spatially concatenate the target and condition images (left-right) at the input level. For generation, the target is paired with a blank image; for editing, it is paired with the source image. To resolve task ambiguity, we prepend explicit task tokens (*i.e.*, [Generate] or [Edit]) to the text prompts. This design enable effective task routing within a shared parameter space without introducing additional parameters or specialized branches.

Training the compact model with a unified scheme is challenging due to its limited capacity and the divergent optimization objectives of generation and

editing tasks. To ensure stability, we propose a task-progressive joint pretraining scheme. Specifically, we introduce an intermediate editing pretraining stage between text-to-image pretraining and joint training. This stage aligns visual condition with the generative latent space prior to complex unified joint optimization, thereby facilitating stable training. Consequently, the pretraining of DreamLite is divided into three progressive stages: *i.e.*, T2I Pretraining $\rightarrow$ Editing Training $\rightarrow$ Unified Joint Training. Following this, we further adopt a two-stage post-training strategy: supervised fine-tuning (SFT) on a curated high-quality dataset, followed by reinforcement learning (RL) for preference alignment. During RL training, we employ HPSv3 [31] as the reward model for generation and EditReward [45] for editing, optimizing the diffusion model via ReFL [48]. This post-training phase consistently enhances both perceptual quality and instruction following, enabling our compact model to outperform prior on-device diffusion baselines.

Overall, DreamLite achieves GenEval (0.72) for text-to-image generation, and ImgEdit (4.11) for text-guided image editing. It outperforms specialized lightweight baselines such as SnapGen, SANA-0.6B and VIBE while remaining competitive with larger unified models such as OmniGen2 and Bagel. These results validate the efficacy of our unified architecture. To minimize deployment overhead, we apply DMD2 [51] to compress the sampling process into 4 denoising steps. On a representative smartphone (*i.e.*, iPhone 17 pro), DreamLite completes a 1024^2 generation or editing task in less than 3 second. To the best of our knowledge, DreamLite is the first unified on-device diffusion model to support both generation and editing within a single network for mobile deployment.

Our contributions are summarized as follows:

- We propose, to the best of our knowledge, the first unified on-device model that supports both text-to-image generation and text-based image editing, eliminating the need to deploy two separate models.
- We introduce an in-context conditioning mechanism for UNet to unify generation and editing, and propose a task-progressive joint pretraining scheme (*i.e.*, T2I $\rightarrow$ Edit $\rightarrow$ Unified Joint Training) to stably train the model.
- DreamLite achieves competitive performance on standard benchmarks and outperforms prior mobile models. After deployment on iPhone 17 pro , DreamLite could generate or edit a 1024×1024 image in less than 3s.

2 Related Work

2.1 Unified Generative Models

Large-scale image models are increasingly built as unified generative systems that support both text-to-image generation and instruction-based editing through a single model. Recent model families such as FLUX 2 [22], HunyuanImage [6], Seedream 4.0 [37], Qwen-Image-2 [43], as well as commercial systems like Gemini-Image [33], GPT-Image [15], LongCat-Image [40] and DeepGen [42], all move toward “generate + edit” as first-class capabilities, typically by strengthening

text-image alignment and instruction following with large backbones and curated post-training. A representative line is FLUX.2, which frames unified generation and editing via an in-context formulation. Compared to these large or cloud-oriented unified systems, our work targets a *single compact* diffusion model that supports both generation and instruction-based editing for *on-device* deployment under strict memory/latency constraints.

2.2 Efficient Diffusion Models

A growing body of work improves diffusion efficiency by optimizing architectures, attention mechanisms, and training/inference recipes. PixArt- Σ [7] explores transformer efficiency for high-resolution generation with key-value compression to alleviate attention cost at large token counts. And SANA [47] proposes linear-attention diffusion transformers to reduce the quadratic complexity of self-attention at high resolutions while maintaining competitive generation performance. Beyond pure generation, EditMGT [8] and VIBE [1] presents a compact instruction-based editing pipeline that combines a lightweight vision-language model with an efficient diffusion backbone, demonstrating that strong editing can be achieved without relying on extremely large diffusion models. Our method is complementary to these efforts: rather than only accelerating T2I or only optimizing editing, we focus on an efficient *unified* interface that consolidates T2I and editing into a single compact model.

2.3 On-Device Generative Models

To enable on-device deployment, prior works have explored quantization, pruning, and knowledge distillation to reduce model size and latency. Early on-device systems [24, 53] pruned and distilled U-Net architectures to generate 512-pixel images within seconds. SnapGen Series [16, 17] have explored efficient diffusion U-Net and transformers tailored for mobile devices and edge deployment. More recently, our concurrent work Mobile-O [38] has demonstrated that a mobile-based diffusion model can perform generation and understanding at the same time. Despite this progress, most on-device works primarily emphasize T2I generation, while instruction-based editing often requires either separate models or additional editing-specific components. Our work targets this gap by providing a unified interface and training strategy so that a single compact model supports both “generate an image” and “edit my photo,” reducing deployment complexity and resource consumption on mobile devices.

2.4 RLHF

Post-training alignment has emerged as a crucial stage for enhancing perceptual quality and instruction compliance, surpassing the limitations of noisy web-scale pre-training. A common practice involves utilizing learned reward models (*e.g.*, ImageReward [48], HPSv2/v3 [31, 46], PickScore [20], and the

editing-specific EditReward [45]) as optimization targets to improve aesthetics and prompt faithfulness while mitigating visual artifacts. On the optimization front, various methods adapt reinforcement learning (RL) to generative models. For instance, ReFL [48] performs reward-guided fine-tuning for diffusion models under constrained backprop settings to reduce training cost, and has inspired follow-up explorations of more efficient preference optimization. Parallel to RL-based approaches, DPO-like objectives (*e.g.*, Diffusion-DPO [41], Align-Prop [34]) have been developed to leverage pairwise preferences without explicit reward modeling. Recent flow-model works also explore GRPO variants (*e.g.*, Flow-GRPO [27], DanceGRPO [49]) for better stability and instruction following under preference feedback.

2.5 Step Distillation

Few-step sampling is essential for interactive and on-device applications. A representative line is consistency-based distillation, including Latent Consistency Models (LCM) [30], which distill time-consistent behavior to enable generation in a small number of steps. Another influential family is distribution matching distillation. DMD [52] distills a multi-step teacher into a few-step student via distribution matching objectives, and DMD2 [51] further improves stability and sample quality under aggressive step reduction. In practice, these methods are often used to compress sampling to $\leq 4 - 8$ steps with acceptable quality, and have been integrated into various backbones (including SD, SDXL and Qwen-Image, *etc.*). Adversarial-based distillation is also a prominent approach; for instance, SDXL-Turbo and SD-Turbo [36] utilize Adversarial Diffusion Distillation (ADD) to achieve high-fidelity, real-time synthesis in a single step. In this work, we employ the DMD2 to compress our sampling process to 4 steps.

3 Proposed Method

This section presents the training pipeline and details of our DreamLite. We first describe the model architecture (Section 3.1), and then detail our training procedure, including task-progressive joint pretraining (Section 3.2), post training (Section 3.3), and few-step distillation (Section 3.4).

3.1 Model Architecture

As illustrated in Fig. 2, the architecture of DreamLite consists of three primary modules: a UNet backbone, a Variational Autoencoder (VAE), and a text encoder. To support unified image generation and editing, we further introduce the in-context conditioning mechanism.

Variational Autoencoder. Following [21,35], we adopt a latent diffusion framework. To enable efficient on-device deployment, we employ an extremely lightweight VAE (*i.e.*, TinyVAE [32]), which contains only 2.5M parameters for image tokenization. It maps image x as a 4-channel latent z with an 8×8 downsampling factor, facilitating efficient training and inference.

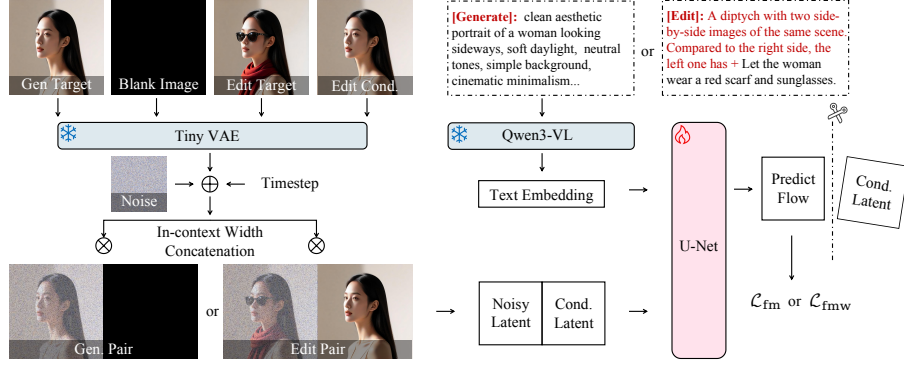


Fig. 2: Overview of DreamLite architecture. It consists of three primary modules: a UNet backbone, a Variational Autoencoder (VAE), and a text encoder. To support unified generation and editing, an in-context conditioning framework is adopted that unifies both tasks at the input level.

Compact UNet. We further develop a compact diffusion model designed for efficient mobile generation. Following SnapGen [16], we adopt the SDXL UNet backbone and optimize it by reducing both its depth and width. To enhance efficiency, we employ several key modifications: (i) removing self-attention layers at high-resolution stages to mitigate quadratic complexity; (ii) replacing standard convolutions with expanded separable convolutions (expansion ratio = 2); and (iii) adopting multi-query attention (MQA) to reduce both computational overhead and memory footprint. More details can be seen in supplementary materials.

Text Encoder. For text conditioning, we utilize Qwen3-VL-2B [3] as our text encoder. We leverage its robust visual-language comprehension capabilities to accurately interpret complex user instructions and process multimodal inputs. This choice ensures precise semantic alignment between the input instructions and the generated content.

In-context Conditioning Mechanism. Existing UNet-based image editing methods [4, 12, 19] typically follow the InstructPix2Pix paradigm [4], which concatenates the condition image with the noisy latent in the channel dimension and fine-tunes the model. However, this mechanism inevitably degrades the generative priors of the pretrained text-to-image (T2I) model and hinders the development of a unified architecture. To address these limitations, we propose extending the UNet with an in-context conditioning framework that unifies both image generation and editing tasks at the input level within a single compact network. As shown in Fig. 2, we construct a two-panel latent by concatenating the latent of the target image (z_{tgt}) and the conditioning image z_{cond} along the width dimension: $z^{\text{pair}} = \text{Concat}(z_{\text{tgt}}, z_{\text{cond}})$. Here, the left panel corresponds to the target output, while the right panel provides the visual condition. This concatenated latent z^{pair} is fed directly into the UNet. For text-to-image generation, we set the conditioning panel to a blank (all-zeros) latent to repre-

sent “no visual condition.” This design allows the model to extend from T2I to editing directly without introducing additional modules, making it highly suitable for a unified framework. To further reduce task ambiguity when training a single model for dual behaviors, we prepend explicit task tokens to the text prompt: **[Generate]** for generation tasks and **[Edit]** for editing tasks. These tokens act as lightweight routing signals without requiring extra parameters or task-specific branches, thereby improving both generation quality and edit controllability under a unified framework. Further details and motivation regarding the architectural design can be found in Section 4.

3.2 Task-progressive Joint Pretraining

Training a compact model with a unified formulation is challenging due to its limited capacity and the divergent optimization objectives of generation and editing tasks. To ensure stable convergence, we propose a task-progressive joint pretraining scheme. Unlike standard approaches that transition directly from text-to-image pretraining to joint training, we introduce an intermediate editing pretraining stage. This stage serves to align the visual conditioning representations with the generative latent space before the complex joint optimization, thereby mitigating task interference. Specifically, the pretraining of DreamLite is divided into three progressive stages: (i) T2I Pretraining, (ii) Editing Training, and (iii) Unified Joint Training.

Text-to-Image Pretraining. We first train DreamLite as a standard text-to-image diffusion model using the flow matching objective [26, 29]. During training, the noisy latent z_t is constructed through linear interpolation between the initial Gaussian noise ϵ and the clean image latent z , *i.e.*, $z_t = F(z, t) = t \cdot z + (1 - t) \cdot \epsilon$ and $t \in [0, 1]$. The model is then trained to predict the velocity of the vector field that defines the trajectory between the noise and data distributions, *i.e.*, $v_t = z - \epsilon$. The training objective is formulated as:

$$\mathcal{L}_{\text{fm}} = \mathbb{E}_{t, z, \epsilon, y} [\|v_\theta(z_t, t, y) - (z - \epsilon)\|^2], \quad (1)$$

where v_θ denotes the denoising UNet and θ represents the learnable parameters. y denotes the conditional embeddings. To improve convergence and stability, we adopt a progressive resolution curriculum following prior work [16]. The training proceeds sequentially from 256×256 to 512×512 , and finally to 1024×1024 resolution, with a multi-scale training strategy applied at each stage. Furthermore, following Stable Diffusion 3 [10], we employ a logit-normal noise sampler to concentrate training on intermediate timesteps. We also utilize dynamic time shifting [21] to scale noise levels according to image resolution. Collectively, this stage establishes a strong generative prior for subsequent training stages.

Editing Training. Following T2I pretraining, we activate the in-context conditioning mechanism and continue training the model on paired text-guided image editing data. The primary objective of this stage is to align the newly introduced visual conditioning with the pre-trained generative latent space, enabling the model to generate images based on both condition images and editing instructions. We also employ a flow-matching loss during this phase. However, a

major challenge in training for editing tasks is that the target edit regions in the target images are usually small. If a standard uniform loss is applied to the entire image, the gradient signal from these small edited regions can be overwhelmed by the dominant signal from the unchanged background, leading to unsatisfactory editing results. Following UniWorld-V1 [25], we adopt a foreground-emphasis mask to re-weight the editing loss. Specifically, we compute a weighting mask w based on the difference between the source and target images, which is then applied to the flow matching objective:

$$\mathcal{L}_{\text{fmw}} = \mathbb{E}_{t,z,\epsilon,y} [\|w \odot (v_{\theta}(z_t, t, y) - w \odot (z - \epsilon))\|^2]. \quad (2)$$

Intuitively, this mask constrains the model to concentrate its capacity on the regions that require change, ensuring high editing fidelity and visual consistency. As shown in Fig. 2, only the outputs corresponding to the target image area are used for loss calculation. More details can be found in supplementary materials. **Unified Joint Training.** Subsequently, we perform unified joint training on a mixture of T2I and editing data. This stage is designed to consolidate the generative priors established in the T2I stage with the instruction-following capabilities acquired during Edit pretraining, ensuring the model converges to a single set of parameters that reliably supports both behaviors. As stated in Sec. 3.1, to mitigate task conflict in this unified setting, we prepend explicit task tokens to the text prompts: **[Generate]** to indicate the absence of a visual reference for generation tasks, and **[Edit]** for editing instructions. These tokens function as lightweight routing signals, guiding the model to switch behaviors dynamically based on the input context. Consequently, DreamLite achieves robust performance in both generation and editing under a strictly unified architecture, eliminating the need for additional parameters. Further experimental analysis of our task-progressive pretraining scheme can be found in Section 4.

3.3 Post Training

While our task-progressive joint pretraining establishes a strong foundation for both generation and editing, the model’s behavior remains unstable due to the high variance of the pretraining data. To improve stability and performance, we employ a two-stage post-training strategy: Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL).

Supervised Fine-Tuning (SFT). The primary goal of SFT is to refine the model’s behavior by exposing it to a curated distribution of high-quality data. To achieve this, we construct a dataset comprising approximately 0.5M samples, selected for their high visual quality and caption diversity. By fine-tuning on these cleaner, denser signals, we effectively steer the model’s target distribution toward a manifold of higher realism and precise instruction following.

Reinforcement Learning (RL). To further align DreamLite with human preferences, we incorporate Reinforcement Learning from Human Feedback (RLHF). Specifically, we adopt the Reward Feedback Learning (ReFL) framework [48], which directly leverages scalar signals from a pre-trained reward model to guide

the denoising trajectory via gradient backpropagation. During training, given a condition c and a timestep t , the model progressively denoises the latent to time t to predict the corresponding clean image $\hat{x}$. A scalar reward $r(c, \hat{x})$ is then computed using the reward model, and gradients are backpropagated through the denoising process. We adopt a ReLU-truncated reward formulation:

$$\mathcal{L}_{\text{r1}} = -\max(0, r(c, \hat{x}) - b), \quad (3)$$

where b is a hyperparameter introduced to ensure training stability and mitigate reward hacking. In our experiments, we employ task-specific reward models to optimize distinct capabilities. For text-to-image generation, we utilize HPSv3 [31], a state-of-the-art preference model built upon the Qwen-VL backbone, with $b = 11$. For image editing, we employ EditReward [45], which is explicitly designed to evaluate adherence to editing instructions, and use $b = 2.5$. This post-training stage enhances both perceptual quality and instruction following, enabling our compact model to outperform prior mobile diffusion baselines.

3.4 Step Distillation

While our post-training strategy ensures high-quality generation, it still requires tens of denoising steps to generate/edit an image, creating a significant bottleneck for real-time mobile applications. To bridge the gap between high-fidelity synthesis and low-latency deployment, we apply Distribution Matching Distillation (DMD) [51, 52] to reduce the sampling process to just 4 steps without compromising visual quality. DMD distills a multi-step diffusion teacher into a few-step generator G by minimizing the approximate Kullback-Leibler (KL) divergence between the distribution of real images $p_{\text{real},t}$ and the generator’s output distribution $p_{\text{fake},t}$. The gradient of DMD loss is computed as:

$$\begin{aligned} \nabla \mathcal{L}_{\text{DMD}} &= \mathbb{E}_t (\nabla_{\theta} \text{KL}(p_{\text{fake},t} \| p_{\text{real},t})) \\ &= -\mathbb{E}_t \left(\int (s_{\text{real}}(F(G_{\theta}(\epsilon), t), t) - s_{\text{fake}}(F(G_{\theta}(\epsilon), t), t)) \frac{dG_{\theta}(\epsilon)}{d\theta} d\epsilon \right), \end{aligned} \quad (4)$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is a random Gaussian noise, and θ denotes the generator parameters. F defines the forward diffusion process that transfers the generated sample $G_{\theta}(\epsilon)$ the noise level corresponding to timestep t . s_{real} and s_{fake} represent the scores, approximated using diffusion models v_{real} and v_{fake} , respectively. To stabilize training, we also employ the GAN loss $\mathcal{L}_{\text{GAN}}$ [51] to enhance the diversity and realism of the generated images for the distilled model. We also adopt a progressive distillation scheme (*i.e.*, T2I $\rightarrow$ Edit $\rightarrow$ Unified) to enable efficient generation and editing in only 4 sampling steps without need of CFG.

4 Experiments

4.1 Implementation Details

DreamLite is implemented using the PyTorch. The training corpus consists of 20M text-to-image pairs and 1.7M image editing samples. During the unified

Table 1: Evaluation on GenEval benchmark. “Param.” denotes backbone parameters.

Model	Param.	Single Obj.	Two Obj.	Counting	Colors	Position	Color	Attr.	Overall
FLUX.1-Dev [21]	12B	0.99	0.81	0.79	0.74	0.20	0.47		0.67
BAGEL [9]	7B	0.99	0.94	0.81	0.88	0.64	0.63		0.82
OmniGen2 [44]	4B	1.00	0.95	0.64	0.88	0.55	0.76		0.80
Z-Image [5]	6B	1.00	0.94	0.78	0.93	0.62	0.77		0.84
LongCat-Image [40]	6B	0.99	0.98	0.86	0.86	0.75	0.73		0.87
DeepGen1.0 [42]	2B	0.99	0.97	0.68	0.90	0.71	0.71		0.83
SANA-1.6B [47]	1.6B	0.99	0.80	0.62	0.89	0.27	0.43		0.67
Hunyuan-DiT [6]	1.5B	0.97	0.77	0.71	0.88	0.13	0.30		0.63
MEISSONIC [2]	1B	0.99	0.66	0.42	0.86	0.10	0.22		0.54
SnapGen [16]	0.38B	-	-	-	-	-	-		0.70
SnapGen++ [17]	0.4B	-	-	-	-	-	-		0.66
SANA-0.6B [47]	0.59B	0.98	0.78	0.59	0.86	0.22	0.40		0.64
Nitro-E-GRPO [39]	0.3B	1.00	0.88	0.63	0.86	0.32	0.48		0.70
Ours	0.39B	0.98	0.88	0.65	0.90	0.35	0.52		0.72

Table 2: Evaluation on ImgEdit benchmark. “Param.” denotes backbone parameters.

Model	Param.	Add	Remove	Replace	Adjust	BG	Style	Extract	Action	Hybrid	Overall
Kontext-Dev [23]	12B	4.04	2.92	4.28	3.89	4.00	4.43	2.71	4.52	3.08	3.76
BAGEL [9]	7B	3.81	3.16	3.85	3.59	3.39	4.51	1.58	4.25	2.67	3.42
OmniGen2 [44]	4B	3.57	3.20	3.74	3.06	3.57	4.81	1.77	4.68	2.52	3.44
Z-Image [5]	6B	4.40	4.13	4.57	4.14	4.14	4.85	4.30	4.50	3.63	4.30
LongCat-Image [40]	6B	4.66	4.68	4.75	4.64	4.35	4.81	3.90	4.83	3.80	4.49
DeepGen1.0 [42]	2B	4.52	4.43	4.42	4.09	4.38	4.48	1.75	4.57	3.61	4.03
VIBE [1]	1.6B	3.89	4.42	4.34	4.22	4.22	4.40	2.90	2.75	3.52	3.85
EditMGT [8]	0.96B	2.78	2.89	3.34	3.29	3.05	4.17	1.56	2.94	2.01	2.89
Ours	0.39B	4.59	4.27	4.32	4.46	4.28	4.42	2.70	4.73	3.24	4.11

joint training stage, we employ a sampling ratio of approximately 1:1 between T2I and editing data to balance performance. We utilize the AdamW optimizer with a staged learning rate schedule: 1×10^{-4} for initial T2I pre-training, 1×10^{-5} for editing training and 1×10^{-6} for unified joint training.

4.2 Quantitative Results

To evaluate the performance of DreamLite, we conduct extensive comparative experiments across multiple mainstream benchmarks, focusing on both Image Editing and Image Generation capabilities.

Performance on Image Generation. We first assess the generative quality using GenEval [13] and DPG [18] as the primary benchmarks with all images generated at 1024×1024 resolution for a fair comparison. The compared baselines are categorized into three groups: (i) **Unified Models** (*e.g.*, BAGEL, OmniGen2), which typically feature backbone sizes exceeding 2B parameters; (ii) **Lightweight Generative Models (<2B)** (*e.g.*, SANA-1.6B, Meisssonic), which remain unsuitable for seamless on-device deployment; and (iii) **On-device**

Table 3: Extended quantitative comparison on DPG and GEdit-EN benchmarks. We report model parameters (Param), DPG scores across six sub-categories and GEdit-EN metrics evaluated by Qwen-VL-2.5. Q_{SC} , Q_{PQ} , and Q_O denote Semantic Consistency, Perceptual Quality, and Overall score, respectively.

Model	Param	DPG							GEdit-EN		
		Global	Entity	Attr.	Rel.	Other	Overall		Q_{SC}	Q_{PQ}	Q_O
FLUX.1/Kontext [21,23]	12B	82.1	89.5	88.7	91.1	89.4	84.0		6.61	7.24	6.79
BAGEL [9]	7B	88.9	90.4	91.3	90.8	88.7	85.1		7.66	7.08	7.20
OmniGen2 [44]	4B	88.8	88.8	90.2	89.4	90.3	83.6		7.11	7.22	6.79
LongCat-Image [40]	6B	89.1	92.5	92.0	93.3	87.5	86.8		7.87	7.49	7.55
DeepGen1.0 [42]	2B	84.1	91.0	88.6	94.0	85.3	84.6		7.83	7.47	7.54
SANA-1.6B [47]	1.6B	86.0	91.5	88.9	91.9	90.7	84.8		-	-	-
VIBE [1]	1.6B	-	-	-	-	-	-		7.61	7.34	7.28
SnapGen [16]	0.38B	-	-	-	-	-	81.1		-	-	-
SnapGen++ [17]	0.4B	-	-	-	-	-	85.2		-	-	-
SANA-0.6B [47]	0.6B	83.0	89.5	89.3	90.1	90.2	83.6		-	-	-
EditMGT [8]	0.96B	-	-	-	-	-	-		6.77	6.74	6.33
Ours	0.39B	84.9	91.6	89.7	94.3	80.4	85.8		7.04	7.54	6.88

Generative Models (<1B) (*e.g.*, SnapGen series, SANA-0.6B), which are restricted to generation-only tasks. Table 1 and Table 3 detail the parameter counts, per-category scores and overall performance on GenEval and DPG benchmark. Note that for the SnapGen series, individual scores are omitted as the models are not open-sourced. Nevertheless, DreamLite (0.38B) achieves competitive performance, rivaling models with nearly 10× more parameters.

Performance on Image Editing. We evaluate editing proficiency on ImgEdit [50] and GEdit [28], using GPT-4o (2024-11-20) and Qwen2.5-VL [3] as the respective automated judges; for GEdit we focus on English tasks. Following the same protocol as the generation task, we group the SOTA baselines by model size into three categories: **Unified Models** (*e.g.*, FLUX and LongCat-Image); **Lightweight Editing Model** (*e.g.*, VIBE); and **On-device Editing Model** (*e.g.*, EditMGT). To our knowledge, DreamLite is the only known on-device editing model with a scale well under 0.5B. Table 2 reports model sizes, per-category scores, and overall ImgEdit performance, where DreamLite achieves SOTA among all lightweight models.

4.3 Qualitative Results

To further validate the visual fidelity and instruction-following capability of our DreamLite, we present a qualitative comparison with both large unified models and specialized lightweight baselines in Fig. 3 and Fig. 4. More results can be found in supplementary materials.

Visual Analysis of Image Generation. As illustrated in Fig. 3, although DreamLite is significantly smaller than other large models, it maintains high structural integrity and semantic alignment across diverse scenarios. In complex

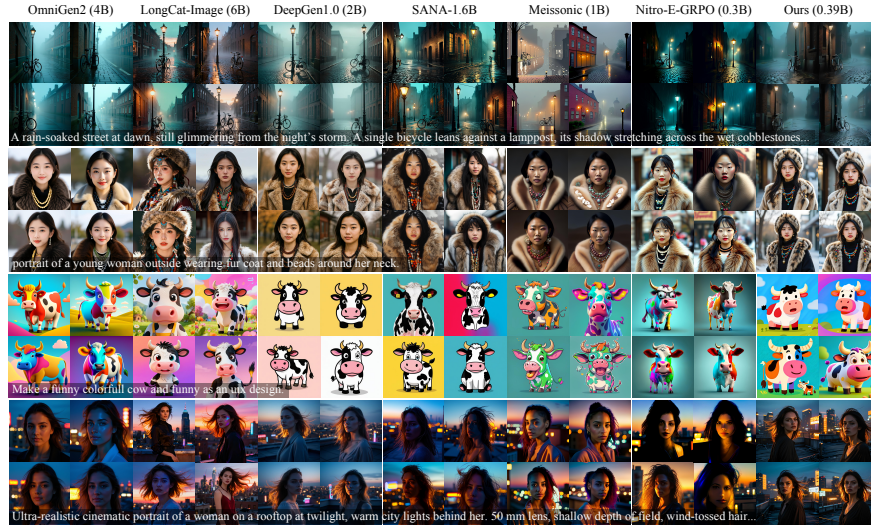


Fig. 3: Comparison of image generation. All input prompts are listed at the bottom of the images.

prompts such as “a rain-soaked street at dawn”, DreamLite produces consistent structural details of “bicycle”. Compared to other lightweight models like Nitro-E-GRPO or Meisssonc, our results exhibit superior composition and fewer artifacts in high-resolution (1024×1024) synthesis.

Visual Analysis of Image Editing. The image editing comparisons in Fig. 4 further highlight the performance of DreamLite. For instance, in the “Raise the person’s right arm” task, DreamLite successfully executes the pose modification while maintaining strict identity and background consistency. This is a significant improvement over specialized models like EditMGT or VIBE, which often suffer from localized distortion or subject identity drift. Furthermore, in object adjusting and removal tasks (*e.g.*, “Place a bicycle in the foreground” or “Remove the white baby onesie”), our model demonstrates a precise understanding of spatial relationships and occlusion comparable to large scale models.

4.4 Ablation Study

To dissect the contribution of our proposed training strategies, we conduct an extensive ablation study on DreamLite, evaluating generation and editing performance on GenEval and ImgEdit benchmarks. The quantitative results are summarized in Table 4 and the qualitative results are shown in Fig. 5.

Condition Mechanism. We first investigate the effectiveness of the In-context (IC) mechanism compared to the traditional Pix2Pix-style channel concatenation after Text-to-image pretraining. As shown in Table 4, when maintaining the same editing training stage (*i.e.*, T2I $\rightarrow$ Edit, row 2&5), the IC mechanism yields a significant improvement from 3.67 to 3.88. Furthermore, compared to

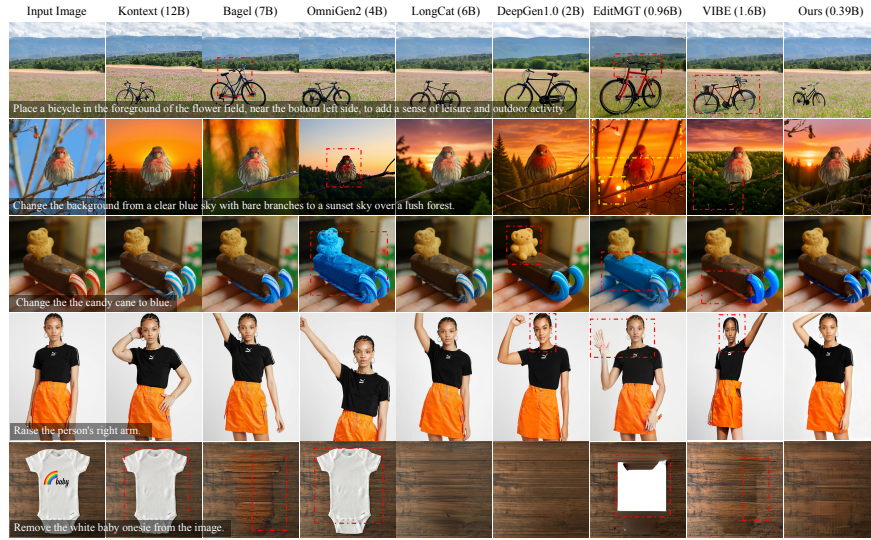


Fig. 4: Comparison of image editing. All source images and instructions are from ImgEdit Benchmark.

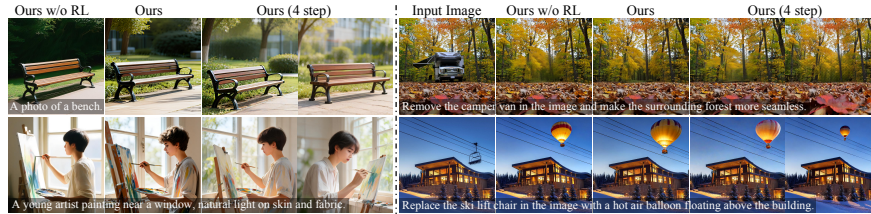


Fig. 5: Ablation study on generation (left) and editing (right) tasks. From left to right in each task: results without RL, our full model, and our model after 4-step distillation.

the results at unified training stage (*i.e.*, $T2I \rightarrow \text{Edit} \rightarrow \text{Unified}$, row 3&7), the IC mechanism proves more suitable for joint training, particularly for the generation task (from 0.61 to 0.71). We attribute this to the fact that IC treats the generation as a special case of in-context referencing (*i.e.*, the condition image is represented by a blank image). This formulation allows the model to leverage a unified spatial prior across disparate behaviors.

Training Recipe. The transition from specialized tasks to unified behavior often encounters a performance bottleneck. Our results indicate that direct joint training (*i.e.*, $T2I \rightarrow \text{Unified}$, row 6) sacrifices both generation and editing precision simultaneously (*e.g.*, GenEval 0.65 and ImgEdit 3.14). By introducing Task-progressive Joint (TPJ) Pre-training (*i.e.*, $T2I \rightarrow \text{Edit} \rightarrow \text{Unified}$, row 7), we observe that the model not only recovers but also surpasses the performance of separate tasks (GenEval 0.71 *vs.* 0.70 and ImgEdit 3.91 *vs.* 3.88). This demonstrates that TPJ allows the compact backbone to internalize the fundamental

Table 4: Ablation study on GenEval and ImgEdit benchmarks. “TPJ” denotes “Task-progressive Joint”.

Exp.	Mechanism	Training Stage	GenEval $\uparrow$	ImgEdit $\uparrow$
Text-to-image Pretraining			0.70	-
Condition Mechanism	Pix2Pix	T2I $\rightarrow$ Edit	0.56	3.67
	Pix2Pix	T2I $\rightarrow$ Edit $\rightarrow$ Unified	0.61	3.65
Training Recipe	In-context	T2I $\rightarrow$ T2I	0.65	-
	In-context	T2I $\rightarrow$ Edit	0.64	3.88
	In-context	T2I $\rightarrow$ Unified	0.65	3.14
	In-context	T2I $\rightarrow$ Edit $\rightarrow$ Unified	0.71	3.94
Reinforcement Learning	In-context	TPJ Pretrain $\rightarrow$ RLHF	0.72	4.11
Step Distillation	In-context	TPJ Pretrain $\rightarrow$ RLHF $\rightarrow$ DMD	0.70	3.8

logic of IC referencing before tackling unified modeling. We also conduct in-context T2I training following standard T2I pre-training (row 4), which results in a slight decline in GenEval performance. We posit that since the blank image provides no information, the model fails to utilize it for contextual referencing, thereby slowing down convergence and compromising T2I performance.

Reinforcement Learning. As shown in Table 4, integrating RLHF into the TPJ pipeline yields the highest overall performance, achieving 4.11 on ImgEdit and 0.72 on GenEval. Beyond these numerical gains, visual evidence in Fig. 5 demonstrates a substantial leap in image aesthetics and high-frequency details. Specifically, it markedly improves background realism in generation tasks and enhances human identity maintenance during editing.

Step Distillation. DMD significantly accelerates the sampling process, albeit with a slight performance penalty in complex semantic alignment tasks. Fig. 5 presents the visual results of DreamLite with 4-step inference after Distribution Matching Distillation. This performance-efficiency trade-off is expected, as the compression of the ODE trajectory inherently constrains the model’s capacity to navigate high-dimensional latent manifolds for intricate edits.

4.5 On-device Deployment

To evaluate the practical utility of DreamLite, we deploy it on both iOS and Android platforms. On iOS, we use an iPhone 17 Pro to obtain a full breakdown of latency and memory. On Android, we use a Xiaomi 14 with the Snapdragon 8 Gen3 Qualcomm NPU and a vivo X100 with the Dimensity 9300 MTK APU. All experiments adopt a 1024×1024 resolution and a 4-step sampling schedule. Visual results on mobile device can be seen in supplementary materials.

Deployment on iOS. Table 5 reports the full latency and memory breakdown on iPhone 17 Pro. We apply 4-bit quantization to the text encoder and keep U-Net and VAE in FP16. The end-to-end generation finishes within ~ 2.7 s and editing within ~ 3.0 s. Peak runtime memory stays at 2.5 GB for generation and 3.5 GB for editing.

Deployment on Android. On Android, we further compress the U-Net with

Table 5: DreamLite generation breakdown on iPhone 17 Pro at 1024×1024 resolution.

Component	Quantization	Load Mem.	Runtime Mem.	Latency [ms]
Qwen3-VL-2B Gen	4-bit	~860 MB	~250 MB	~300
Qwen3-VL-2B Edit	4-bit	~860 MB	~810 MB	~800
U-Net $\times 4$ steps	FP16	~125 MB	~1320 MB	~580 $\times 4$
VAE Encoder Edit	FP16	~2 MB	~4 MB	~70
VAE Decoder	FP16	~2 MB	~6 MB	~50
Total-Gen Peak	–	1.0 GB	2.5 GB	~2700
Total-Edit Peak	–	1.0 GB	3.5 GB	~3000

Table 6: On-device inference latency [ms] of DreamLite components on Android at 1024×1024 resolution. The U-Net is quantized to W8A8 and the VAE is kept in FP16.

Component	Quantization	Load Mem.	Snapdragon 8 Gen3	Dimensity 9300
U-Net	W8A8	389.00 MB	103.84 $\times 4$	122.53 $\times 4$
VAE Decoder	FP16	2.45 MB	22.17	26.16

W8A8 quantization to fit the NPU and APU budget. As summarized in Table 6, the U-Net costs only 103.84ms per step on Snapdragon 8 Gen3 and 122.53ms on Dimensity 9300. Including VAE decoding around 22ms and system overhead, the end-to-end user experience stays near the 1s threshold.

Analysis. On iPhone 17 Pro the FP16 U-Net dominates the budget at ~580ms per step (>85% of end-to-end cost), while the 4-bit text encoder is reduced to ~300ms. On Android, W8A8 cuts the U-Net to ~104ms and ~123ms per step on Snapdragon 8 Gen3 and Dimensity 9300, a $\sim 5\times$ speedup over iOS FP16. The total U-Net latency over 4 steps reaches 0.42s and 0.49s respectively. Including VAE decoding around 22ms and system overhead, the end-to-end user experience stays near the 1s threshold.

5 Conclusion

We presented DreamLite, a compact unified on-device diffusion model that supports both text-to-image generation and text-guided image editing within a single network. By unifying conditioning through in-context spatial concatenation and explicit task tokens, DreamLite achieves seamless multi-tasking without parameter overhead. To stably train this small-capacity unified model, we introduced a task-progressive curriculum (T2I $\rightarrow$ Edit $\rightarrow$ Joint) followed by high-quality SFT and reinforcement learning. Finally, step distillation compresses sampling to 4 denoising steps for efficient mobile inference. Experiments show that DreamLite (GenEval 0.72, ImgEdit 4.11) significantly outperforms prior on-device baselines and remains competitive with much larger server-side model. On representative smartphone (*i.e.*, iPhone 17 pro), DreamLite could generate or edit a 1024×1024 image in less than 3s.

References

1. Alekseenko, G., Gordeev, A., Tolstykh, I., Suleimanov, B., Dokholyan, V., Fedorov, G., Yakubson, S., Tsybina, A., Chernyshov, M., Kuprashevich, M.: Vibe: Visual instruction based editor. arXiv preprint arXiv:2601.02242 (2026)
2. Bai, J., Ye, T., Chow, W., Song, E., Chen, Q.G., Li, X., Dong, Z., Zhu, L., Yan, S.: Meissonic: Revitalizing masked generative transformers for efficient high-resolution text-to-image synthesis. In: The Thirteenth International Conference on Learning Representations (2024)
3. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
4. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18392–18402 (2023)
5. Cai, H., Cao, S., Du, R., Gao, P., Hoi, S., Hou, Z., Huang, S., Jiang, D., Jin, X., Li, L., et al.: Z-image: An efficient image generation foundation model with single-stream diffusion transformer. arXiv preprint arXiv:2511.22699 (2025)
6. Cao, S., Chen, H., Chen, P., Cheng, Y., Cui, Y., Deng, X., Dong, Y., Gong, K., Gu, T., Gu, X., et al.: Hunyuanimage 3.0 technical report. arXiv preprint arXiv:2509.23951 (2025)
7. Chen, J., Ge, C., Xie, E., Wu, Y., Yao, L., Ren, X., Wang, Z., Luo, P., Lu, H., Li, Z.: Pixart- σ : Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In: European Conference on Computer Vision. pp. 74–91. Springer (2024)
8. Chow, W., Li, L., Kong, L., Li, Z., Xu, Q., Song, H., Ye, T., Wang, X., Bai, J., Xu, S., et al.: Editmgt: Unleashing potentials of masked generative transformers in image editing. arXiv preprint arXiv:2512.11715 (2025)
9. Deng, C., Zhu, D., Li, K., Gou, C., Li, F., Wang, Z., Zhong, S., Yu, W., Nie, X., Song, Z., Shi, G., Fan, H.: Emerging properties in unified multimodal pretraining. arXiv preprint arXiv:2505.14683 (2025)
10. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024)
11. Gao, Y., Gong, L., Guo, Q., Hou, X., Lai, Z., Li, F., Li, L., Lian, X., Liao, C., Liu, L., et al.: Seedream 3.0 technical report. arXiv preprint arXiv:2504.11346 (2025)
12. Geng, Z., Yang, B., Hang, T., Li, C., Gu, S., Zhang, T., Bao, J., Zhang, Z., Li, H., Hu, H., et al.: Instructdiffusion: A generalist modeling interface for vision tasks. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp. 12709–12720 (2024)
13. Ghosh, D., Hajishirzi, H., Schmidt, L.: Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems* **36**, 52132–52152 (2023)
14. Gong, L., Hou, X., Li, F., Li, L., Lian, X., Liu, F., Liu, L., Liu, W., Lu, W., Shi, Y., et al.: Seedream 2.0: A native chinese-english bilingual image generation foundation model. arXiv preprint arXiv:2503.07703 (2025)
15. Google: Gemini 2.5 flash. <https://aistudio.google.com/models/gemini-2-5-flash-image> (2025)

16. Hu, D., Chen, J., Huang, X., Coskun, H., Sahni, A., Gupta, A., Goyal, A., Lahiri, D., Singh, R., Idelbayev, Y., et al.: Snapgen: Taming high-resolution text-to-image models for mobile devices with efficient architectures and training. arXiv preprint arXiv:2412.09619 (2024)
17. Hu, D., Gupta, A., Gabidolla, M., Sahni, A., Coskun, H., Li, Y., Idelbayev, Y., Mahmood, A., Lebedev, A., Lahiri, D., et al.: Snapgen++: Unleashing diffusion transformers for efficient high-fidelity image generation on edge devices. arXiv preprint arXiv:2601.08303 (2026)
18. Hu, X., Wang, R., Fang, Y., Fu, B., Cheng, P., Yu, G.: Ella: Equip diffusion models with llm for enhanced semantic alignment. arXiv preprint arXiv:2403.05135 (2024)
19. Huang, Y., Xie, L., Wang, X., Yuan, Z., Cun, X., Ge, Y., Zhou, J., Dong, C., Huang, R., Zhang, R., et al.: Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8362–8371 (2024)
20. Kirstain, Y., Polyak, A., Singer, U., Matiana, S., Penna, J., Levy, O.: Pick-a-pic: An open dataset of user preferences for text-to-image generation. Advances in neural information processing systems **36**, 36652–36663 (2023)
21. Labs, B.F.: Flux. <https://github.com/black-forest-labs/flux> (2024)
22. Labs, B.F.: Flux2. <https://github.com/black-forest-labs/flux2> (2025)
23. Labs, B.F., Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., et al.: Flux. 1 kontekst: Flow matching for in-context image generation and editing in latent space. arXiv preprint arXiv:2506.15742 (2025)
24. Li, Y., Wang, H., Jin, Q., Hu, J., Chemerys, P., Fu, Y., Wang, Y., Tulyakov, S., Ren, J.: Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. Advances in Neural Information Processing Systems **36**, 20662–20678 (2023)
25. Lin, B., Li, Z., Cheng, X., Niu, Y., Ye, Y., He, X., Yuan, S., Yu, W., Wang, S., Ge, Y., et al.: Uniworld-v1: High-resolution semantic encoders for unified visual understanding and generation. arXiv preprint arXiv:2506.03147 (2025)
26. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
27. Liu, J., Liu, G., Liang, J., Li, Y., Liu, J., Wang, X., Wan, P., Zhang, D., Ouyang, W.: Flow-grpo: Training flow matching models via online rl. arXiv preprint arXiv:2505.05470 (2025)
28. Liu, S., Han, Y., Xing, P., Yin, F., Wang, R., Cheng, W., Liao, J., Wang, Y., Fu, H., Han, C., Li, G., Peng, Y., Sun, Q., Wu, J., Cai, Y., Ge, Z., Ming, R., Xia, L., Zeng, X., Zhu, Y., Jiao, B., Zhang, X., Yu, G., Jiang, D.: Step1x-edit: A practical framework for general image editing. arXiv preprint arXiv:2504.17761 (2025)
29. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003 (2022)
30. Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H.: Latent consistency models: Synthesizing high-resolution images with few-step inference. arXiv preprint arXiv:2310.04378 (2023)
31. Ma, Y., Wu, X., Sun, K., Li, H.: Hpsv3: Towards wide-spectrum human preference score. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15086–15095 (2025)
32. Madebyollin: taesd. <https://github.com/madebyollin/taesd> (2024)
33. OpenAI: Gpt-image. <https://openai.com/index/gpt-4o-system-card/> (2025)
34. Prabhudesai, M., Goyal, A., Pathak, D., Fragkiadaki, K.: Aligning text-to-image diffusion models with reward backpropagation (2023)

35. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 10684–10695 (2022)
36. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation. In: *European Conference on Computer Vision*. pp. 87–103. Springer (2024)
37. Seedream, T., Chen, Y., Gao, Y., Gong, L., Guo, M., Guo, Q., Guo, Z., Hou, X., Huang, W., Huang, Y., et al.: Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427* (2025)
38. Shaker, A., Heakl, A., Muhammad, J., Thawkar, R., Thawakar, O., Li, S., Cholakal, H., Reid, I., Xing, E.P., Khan, S., et al.: Mobile-o: Unified multimodal understanding and generation on mobile device. *arXiv preprint arXiv:2602.20161* (2026)
39. Shen, T., Yu, J., Zhou, D., Li, D., Barsoum, E.: E-mmdit: Revisiting multimodal diffusion transformer design for fast image synthesis under limited resources. *arXiv preprint arXiv:2510.27135* (2025)
40. Team, M.L., Ma, H., Tan, H., Huang, J., Wu, J., He, J.Y., Gao, L., Xiao, S., Wei, X., Ma, X., et al.: Longcat-image technical report. *arXiv preprint arXiv:2512.07584* (2025)
41. Wallace, B., Dang, M., Rafailov, R., Zhou, L., Lou, A., Purushwalkam, S., Ermon, S., Xiong, C., Joty, S., Naik, N.: Diffusion model alignment using direct preference optimization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 8228–8238 (2024)
42. Wang, D., Li, R., Han, F., Ma, C., Song, W., Wang, S., Wang, Y., Xin, Y., Liu, H., Zhang, Z., et al.: Deepgen 1.0: A lightweight unified multimodal model for advancing image generation and editing. *arXiv preprint arXiv:2602.12205* (2026)
43. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-image technical report. *arXiv preprint arXiv:2508.02324* (2025)
44. Wu, C., Zheng, P., Yan, R., Xiao, S., Luo, X., Wang, Y., Li, W., Jiang, X., Liu, Y., Zhou, J., et al.: Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871* (2025)
45. Wu, K., Jiang, S., Ku, M., Nie, P., Liu, M., Chen, W.: Editreward: A human-aligned reward model for instruction-guided image editing. *arXiv preprint arXiv:2509.26346* (2025)
46. Wu, X., Hao, Y., Sun, K., Chen, Y., Zhu, F., Zhao, R., Li, H.: Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341* (2023)
47. Xie, E., Chen, J., Chen, J., Cai, H., Tang, H., Lin, Y., Zhang, Z., Li, M., Zhu, L., Lu, Y., et al.: Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629* (2024)
48. Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., Dong, Y.: Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems* **36**, 15903–15935 (2023)
49. Xue, Z., Wu, J., Gao, Y., Kong, F., Zhu, L., Chen, M., Liu, Z., Liu, W., Guo, Q., Huang, W., et al.: Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818* (2025)
50. Ye, Y., He, X., Li, Z., Lin, B., Yuan, S., Yan, Z., Hou, B., Yuan, L.: Imgedit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275* (2025)
51. Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., Freeman, B.: Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems* **37**, 47455–47487 (2024)

52. Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T., Park, T.: One-step diffusion with distribution matching distillation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6613–6623 (2024)
53. Zhao, Y., Xu, Y., Xiao, Z., Jia, H., Hou, T.: Mobilediffusion: Instant text-to-image generation on mobile devices. In: European Conference on Computer Vision. pp. 225–242. Springer (2024)