NEP: Autoregressive Image Editing via Next Editing Token Prediction

Huimin Wu[1] **Xiaojian Ma**[1] **Haozhe Zhao**[2] **Yanpeng Zhao**[1] **Qing Li**[1]✉
[1]State Key Laboratory of General Artificial Intelligence, BIGAI [2]Peking University
Project website: nep-bigai.github.io

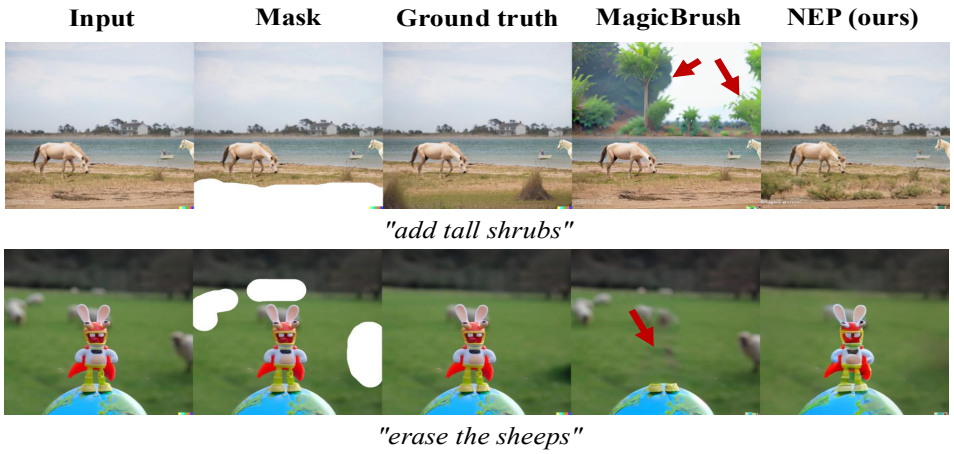

Figure 1: Our approach avoids full-image generation and does not introduce unintended changes as the previous diffusion-model-based editing approach [58].

Abstract

Text-guided image editing involves modifying a source image based on a language instruction and, typically, requires changes to only small local regions. However, existing approaches generate the entire target image rather than selectively regenerate only the intended editing areas. This results in (1) unnecessary computational costs and (2) a bias toward reconstructing non-editing regions, which compromises the quality of the intended edits. To resolve these limitations, we propose to formulate image editing as **N**ext **E**diting-token **P**rediction (NEP) based on autoregressive image generation, where only regions that need to be edited are regenerated, thus avoiding unintended modification to the non-editing areas. To enable any-region editing, we propose to pre-train an any-order autoregressive text-to-image (T2I) model. Once trained, it is capable of zero-shot image editing and can be easily adapted to NEP for image editing, which achieves a new state-of-the-art on widely used image editing benchmarks. Moreover, our model naturally supports test-time scaling (TTS) through iteratively refining its generation in a zero-shot manner.

1 Introduction

Text-driven image editing aims to modify a source image following a given language instruction. Typically, modifications are confined to small local regions (editing regions) while most of the image remains unchanged (non-editing regions). A predominant paradigm for solving the task is through

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

https://doi.org/10.52202/085713-4358

the diffusion model [40, 39, 31], but standard diffusion models struggle with controllable editing, that is, editing only a target region without altering the surrounding areas. To tackle this challenge, an inversion technique has been proposed and augmented with diffusion-based image generation models [39, 21]. The core idea of this method is to find the mapping of non-editing regions to the corresponding subspace of Gaussian noise. It requires that the initial Gaussian noise that can be decoded into the source image should be pre-defined, which is, however, hard to obtain exactly, and further leads to unintended edits [11].

A more controllable paradigm is to pre-define editing regions and edit only the specified areas while preserving the rest [1, 23]. However, these approaches perform full generation of the target image, including regions that are not required to be edited, and thus are suboptimal in terms of efficiency. This inefficiency is pronounced in training-based editing approaches [2, 50], which also demand significant computational resources to learn to reconstruct. Moreover, the reliance on full-image generation introduces a learning bias during training; that is, image editing models tend to prioritize reconstruction for the non-editing regions over regeneration for the intended editing regions [50].

To address these issues, we introduce **N**ext **E**diting-token **P**rediction (**NEP**), a new formulation of text-guided image editing based on autoregressive (AR) image generation. NEP primarily focuses on regeneration for the editing region and removes the need for optimizing reconstruction for the non-editing areas. Consequently, it improves efficiency and circumvents the learning bias simultaneously. Since the standard AR model employs a fixed raster-scan generation order, it is incompatible with NEP's requirement to generate arbitrary editing regions. To address this, we develop NEP using a two-stage training strategy. First, we pre-train RLlamaGen, a robust random-order AR-based text-to-image (T2I) model that supports arbitrary-order generation and zero-shot local editing. In the second stage, we fine-tune RLlamaGen to optimize NEP's editing performance. Additionally, NEP enables test-time scaling through iterative refinement, improving generation outcomes. We summarize our contributions as follows:

- We propose a new formulation of image editing as next editing-token prediction. It simplifies the learning objectives to regeneration only, leading to higher efficiency and better editing quality. Our approach sets up new records on region-based editing tasks and achieves competitive results on free-form editing benchmarks.
- We propose a two-stage training regime for NEP, where the first stage creates RLlamaGen, a new T2I model capable of arbitrary-order full image generation and zero-shot local editing.
- We analyze the test-time scaling behaviors by embedding NEP in an iterative refinement loop.

2 Methods

In this section, we first introduce the pre-training approach RLlamaGen that can generate image tokens in any user-specified order (§2.1). Then, we elaborate on NEP for image editing (§2.2). Finally, we introduce test-time scaling strategies (§2.3) by integrating NEP in an iterative refinement loop.

2.1 NEP Pre-training

Preliminaries on LlamaGen. The NEP framework is versatile and compatible with various design choices[42, 49, 48]. In this work, we build upon LlamaGen[42], the first open-source text-conditioned autoregressive model to outperform diffusion models, leveraging its robust architecture to enable NEP's random-order generation and iterative refinement for enhanced image editing and generation. To maintain potential unification with text modality, the architecture design of LlamaGen largely follows one of the popular LLMs, Llama [45, 46]. The conditioning text embeddings are extracted from FLAN T5 [7], followed by a projector for dimensionality alignment. The text embeddings are left-padded to a fixed length L_T and prefilled to generate image tokens. Images are firstly tokenized by the encoder and quantizer of VQGAN [8], and generated token ids are mapped to RGB pixels by the decoder. Image tokens with length L are generated in a next-token prediction fashion. Formally, given a text sequence T, the sequentialized image tokens $I = \{I_1, I_2, ..., I_L\}$, are generated by:

$$p(I) = \prod_{i=1}^{L} p(I_i | I_{1,...,i-1}; T) \tag{1}$$

https://doi.org/10.52202/085713-4358

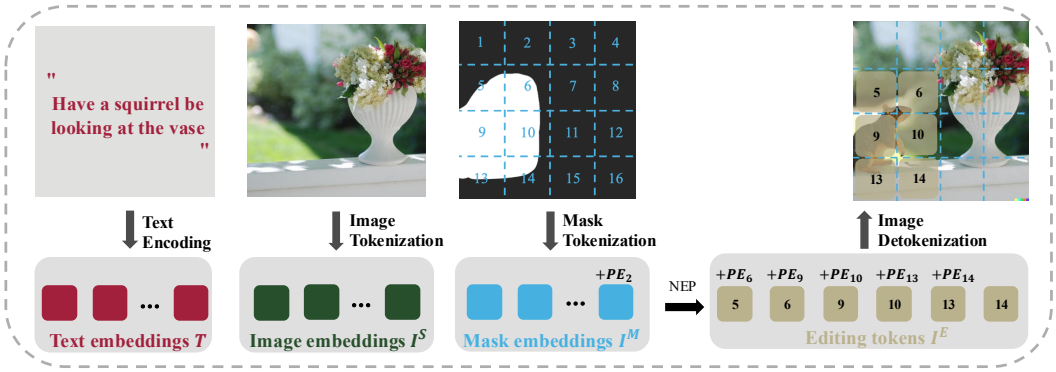

Figure 2: **Overview of Next-Editing-token Prediction.** The input sequence is comprised of: 1) **text embeddings**, extracted from FLAN-T5, 2) **source image embeddings**, tokenized by VQGAN, and 3) **mask embeddings**, a sequence of interleaved editing and non-editing embeddings. The output **editing tokens** (in raster scan order) are filled back to the source image based on the editing mask. PE_i denotes the learned positional embeddings that specify the token generation order.

RLlamaGen: Randomized Autoregressive Text-to-Image Generation. To address LlamaGen's limitation of generating image tokens solely in raster scan order, we extend it to create RllamaGen, which supports generating image tokens in any user-specified order, enabling flexible, arbitrary-order generation [26, 55, 25]. To add order awareness to the model, following [26, 55], we learn an extra sequence of positional embeddings $PE_1, PE_2, ..., PE_L$, which is shuffled based on a random order to define the generation sequence. For each input image token, the positional embedding corresponding to the next token in the assigned order is added. Formally, the generation of an image sequence I_O in the order of $O = [o_1, o_2, ...o_L]$ is defined as:

$$p(I_O) = \prod_{i=1}^{L} p(I_{o_i} | I_{o_1} + PE_{o_2}, ..., I_{o_{i-1}} + PE_{o_i}; T) \tag{2}$$

RLlamaGen supports zero-shot editing by regenerating tokens at given positions, allowing seamless transferability to image editing.

2.2 NEP: Next-Editing-token Prediction

NEP leverages three types of conditioning for region-based editing: 1) text instructions tokens, 2) source images tokens, and 3) editing region masks tokens. The tokenization of text instructions and images remains consistent with the pre-training stage. We detail the construction of editing region conditioning sequences derived from a pixel-level mask $M \in \{0,1\}^{H \times W}$.

Editing Region Conditioning (ERC) We firstly patchify the pixel-level editing mask M by max-pooling each non-overlapping sliding window with the size of $p \times p$. Subsequently, we flatten the patched mask into a sequence $M^E = \{m_1, m_2, ...m_L\} \in \{0,1\}^L$. The masking sequence $I^M = \{I_1^M, ..., I_L^M\}$ is tokenized by querying a two-sized codebook comprising an editing embedding E_{emb} and a non-editing embedding U_{emb}, which is formally defined as:

$$I_i^M = \begin{cases} E_{\text{emb}} & \text{if } M_i^E = 1 \\ U_{\text{emb}} & \text{otherwise} \end{cases} \tag{3}$$

Our editing model processes $L_T + 2 \times L$ input tokens and generates L_E editing tokens, corresponding to the masked target image tokens, denoted as I_E. The generation order corresponds to the positions of the editing tokens within the raster scan order, denoted as $O^E = \{o_1^E, ..., o_{L_E}^E\}$.

Formally, our NEP strategy is defined as:

$$p(I_E) = \prod_{i=1}^{L_E} p(I_{o_i^E} | I_{o_1^E} + PE_{o_2^E}, ..., I_{o_{i-1}^E} + PE_{o_i^E}; T, I^S_{1,...,L}, I^M_{1,...,L}) \tag{4}$$

https://doi.org/10.52202/085713-4358

In scenarios where a region editing mask is unavailable or for global editing tasks (e.g., style transfer), the editing tokens are predicted according to the raster scan order.

2.3 Test-time Scaling with NEP

NEP can be employed to support test-time scaling by integrating it into a self-improving loop. In each refinement step, prior to NEP, a revision region is proposed. Existing image reward models [51] usually produce a single value for the full image. To obtain token-level dense quality scores, we calculate Grad-CAM [35] value regarding the critic model (*i.e.,* off-the-shelf CLIP-ViT-B/32). These values reflect each token's contribution to the overall image quality score, measured by a reward model (*i.e.,* ImageReward [51]). Positions that correspond to the K lowest scores are identified as the revision regions. During revision, we adopt NEP to regenerate tokens in this region, conditioning them on the remaining high-quality tokens. After NEP, the reward model evaluates whether the revised image surpasses the original, determining whether to accept or reject the revision. To further improve quality, for NEP, we apply a rejection sampling strategy, regenerating tokens at the revision positions in multiple random orders and selecting the revision with the highest quality score. This approach demonstrates strong scaling potential, suggesting that effective revision of initial generations can significantly enhance performance.

3 Experiments

We evaluate our framework on the image editing and text-to-image generation tasks. Firstly, we introduce the full training setup that trains the RLlamaGen and NEP stage-by-stage (§3.1). Secondly, we evaluate NEP for image editing and validate its design choices from various aspects (§3.2). Then, we demonstrate the results of NEP pre-training model RLlamaGen (§3.3). Finally, we showcase the test-time scaling behaviors (§3.4).

3.1 Datasets and Training settings

T2I pre-training settings. We use LlamaGen-XL with 775M parameters as the base T2I model and adapt it to RLlamaGen adding 0.3M positional embedding parameters. Our training data consists of around 16M text-image pairs and is collected from multiple open-source datasets, including ALLaVA-LAION [5], CC12M [4], Kosmos-G [24], LAION-LVIS-220 [34], LAION-COCO-AESTHETIC [18], LAION-COCO-17M [56], and ShareGPT4V [6]. We train RLlamaGen for $60,000$ steps with a batch size of 360 and an image resolution of 256×256. The optimizer is Fused AdamW with β_1, β_2 set to 0.9, 0.95, respectively, and a constant learning rate of 1e-4 is used. We perform training on 8 NVIDIA Tesla A100 GPUs, which takes 39 hours.

Image Editing Training Settings. We fine-tune RLlamaGen for image editing by adding two learnable embeddings (i.e., E_{emb} and U_{emb}) to specify masking regions. This strategy is computationally efficient, with only 3.6k parameters introduced. Our editing model is trained on the UltraEdit dataset [60] that comprises 4 million image pairs, where 131k samples are annotated with editing regions. For those with no editing region annotations, we use them for full-image generation.

We perform training on 4 NVIDIA Tesla A100 GPUs. The model is trained for $3.9M$ steps with a batch size of 100 and a learning rate of $1e-4$. Per common practices [58, 60], we evaluate models at a higher image resolution than that used during training (specifically, 512×512 pixels compared to 256×256 pixels), and fine-tune them on the target resolution for an additional $2,000$ steps. For the Emu Edit benchmark, we train our model with a learning rate of $1e-5$ for $60,000$ steps.

3.2 Results on Image Editing

Benchmarks & Evaluation Metrics. We demonstrate the superiority of our approach on two widely recognized benchmarks: MagicBrush [58] and Emu Edit [36]. The MagicBrush test set provides editing region annotations for each sample, thereby facilitating the evaluation of region-conditioned editing. This benchmark assesses both multi-turn editing, which evaluates the final image after a series of edits, and single-turn editing, which assesses the target image following an individual edit.

The MagicBrush benchmark provides target images and evaluates the similarity between each generated image and the corresponding target image using various metrics, including L1 distance, L2

Table 1: **Results on the MagicBrush test set for region-aware editing.** We compare NEP with existing approaches under single-turn and multi-turn settings with our results labeled in gray.

Settings	Methods	L1↓	L2↓	CLIP-I↑	DINO↑
Single-turn	***Global Description-guided***				
	SD-SDEdit	0.1014	0.0278	0.8526	0.7726
	Null Text Inversion	0.0749	0.0197	0.8827	0.8206
	GLIDE	3.4973	115.8347	0.9487	0.9206
	Blended Diffusion	3.5631	119.2813	0.9291	0.8644
	Instruction-guided				
	HIVE	0.1092	0.0380	0.8519	0.7500
	InstructPix2Pix (IP2P)	0.1141	0.0371	0.8512	0.7437
	IP2P w/ MagicBrush	0.0625	0.0203	0.9332	0.8987
	UltraEdit	0.0575	0.0172	0.9307	0.8982
	FireEdit	0.0701	0.0238	0.9131	0.8619
	AnySD	0.1114	0.0439	0.8676	0.7680
	EditAR	0.1028	0.0285	0.8679	0.8042
	Ours	**0.0547**	**0.0163**	**0.9350**	**0.9044**
Multi-turn	***Global Description-guided***				
	SD-SDEdit	0.1616	0.0602	0.7933	0.6212
	Null Text Inversion	0.1057	0.0335	0.8468	0.7529
	GLIDE	11.7487	1079.5997	0.9094	0.8494
	Blended Diffusion	14.5439	1510.2271	0.8782	0.7690
	Instruction-guided				
	HIVE	0.1521	0.0557	0.8004	0.6463
	InstructPix2Pix (IP2P)	0.1345	0.0460	0.8304	0.7018
	IP2P w/ MagicBrush	0.0964	0.0353	0.8924	0.8273
	UltraEdit	0.0745	**0.0236**	0.9045	0.8505
	FireEdit	0.0911	0.0326	0.8819	0.8010
	AnySD	0.0748	0.0273	**0.9152**	**0.8623**
	EditAR	0.1341	0.0433	0.8256	0.7200
	Ours	**0.0707**	0.0269	0.9107	0.8493

distance, CLIP feature similarity (CLIP-I), and DINO feature similarity. Additionally, it measures text-image consistency by comparing the CLIP feature similarity (CLIP-T) between the generated image and the caption of the target image.

The Emu Edit test set does not provide target images; therefore, the evaluation of editing region regeneration is conducted separately from the reconstruction of unedited regions. The regeneration process is assessed using two metrics: CLIP text-image similarity (CLIPout) and CLIP text-image direction similarity (CLIPdir) measure the consistency between the change in images and the change in captions. The reconstruction quality is measured by comparing the edited image to the original source image in terms of L1 distance, CLIP image similarity (CLIPimg), and DINO similarity.

3.2.1 Quantitative Results

We demonstrate the superiority of NEP in terms of region-aware editing on the MagicBrush test set. The compared prior arts broadly fall into two categories: (1) global description-based, such as SD-SDEdit [20], Null Text Inversion [21], GLIDE [23], as well as Blended Diffusion [1], and (2) instruction-guided, including HIVE [59], InstructPix2Pix [2], MagicBrush [58], UltraEdit [60], FireEdit [61], AnySD [54] and EditAR [22]. Table 1 demonstrates that our approach achieves the highest score for single-turn editing and better or comparable performance under the multi-turn setting. For the first time, autoregressive models can achieve top performance on well-recognized editing benchmarks.

We demonstrate the effectiveness of free-form editing on the Emu Edit test set [36]. We compare NEP with state-of-the-art approaches including InstructPix2Pix [2], MagicBrush [58], Emu Edit [36] UltraEdit [60], MIGE [44], and AnySD [54]. In Table 2, we can observe that, without resorting to editing masks, our approach still achieves comparable or better editing performance.

https://doi.org/10.52202/085713-4358

Table 2: **Results on Emu Edit Test for free-form editing.** Our approach is highlighted in gray.

Method	CLIPdir↑	CLIPout↑	L1↓	CLIPimg↑	DINO↑
InstructPix2Pix	0.0784	0.2742	0.1213	0.8518	0.7656
MagicBrush	0.0658	0.2763	0.0652	0.9179	**0.8924**
Emu Edit	0.1066	0.2843	0.0895	0.8622	0.8358
UltraEdit	**0.1076**	0.2832	0.0713	0.8446	0.7937
MIGE	0.1070	0.3067	0.0865	0.8714	0.8432
AnyEdit	0.0626	0.2943	**0.0673**	**0.9202**	**0.8919**
Ours	0.1064	**0.3078**	0.0781	0.8710	0.8440

Table 3: **Ablation studies on the MagicBrush test set under the multi-turn setting**. We validate the contribution of each design choice by removing them and observing the performance drop. We ablate two aspects: 1) **ERC** by removing the editing & unediting tokens inferred from editing region masks, and 2) **NEP vs. NTP** by generating full image tokens. The default setting is highlighted in gray.

Methods	#Output Tokens	L1↓	L2↓	CLIP-I↑	DINO↑
NEP	L_E	**0.0712**	**0.0272**	**0.9097**	**0.8459**
w/o ERC	L_E	0.0741	0.0281	0.9040	0.8372
NTP	L	0.0968	0.0309	0.8854	0.8235

3.2.2 Ablation Study

We perform ablation studies on the Magicbrush multi-turn test set. For each configuration, we report the results of the models trained for $30,000$ steps. We assess two critical design choices. First, we exclude mask embeddings, relying solely on text and source images as inputs, which degrades performance as shown in Table 3. Qualitatively, we observe that removing ERC increases the likelihood of the model making no changes to the source model, as demonstrated in Figure 3. Second, we remove the next editing token positions by generating all tokens in a raster scan order, following an NTP framework. Without any priors on editing regions, this leads to a significant performance drop, highlighting the need for targeted token generation.

3.2.3 Computational efficiency

Table 4 demonstrates comparative results on computational cost. NEP requires higher GPU resources due to the concatenation of mask embeddings along the sequential dimension (Section 2.2), which increases sequence length and attention computational cost. Despite this, our approach achieves the fastest editing speed as we only need to predict editing region tokens rather than the whole image as diffusion models or AR-based models do.

Table 4: Computational cost averaged across MagicBrush test samples.

Methods	Memory (GB)	Inference time (s)
UltraEdit	**4.04**	2.94
EditAR	6.59	10.70
NEP	13.25	**2.88**

3.2.4 Qualitative Results

Figure 4 presents qualitative comparisons with state-of-the-art methods. Apart from avoiding unintended modifications to the input image, as shown in Figure 1, our approach excels in following the provided instructions to perform faithful and accurate modifications. Additionally, it is capable of making fine-grained modifications (e.g., changing the outfit), showcasing its high versatility and precision in handling complex editing instructions.

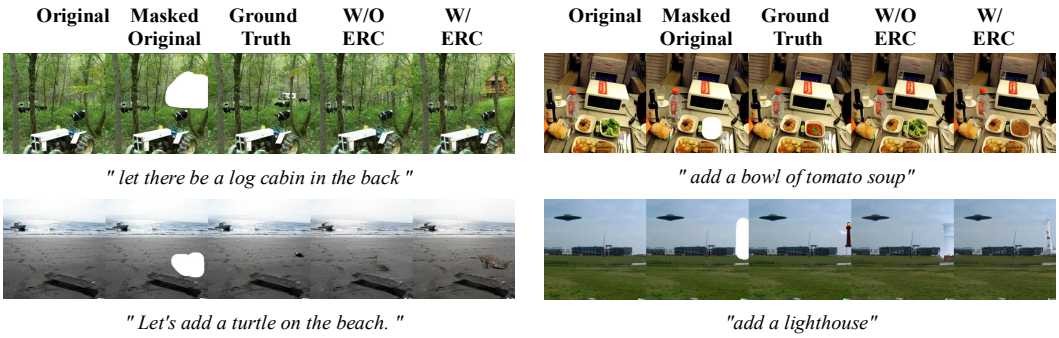

Figure 3: **Visualized ablation on ERC.** This demonstrates that removing Editing Region Conditioning increases the editing model's change to refuse to modify the source image. Best viewed zoomed in and in color.

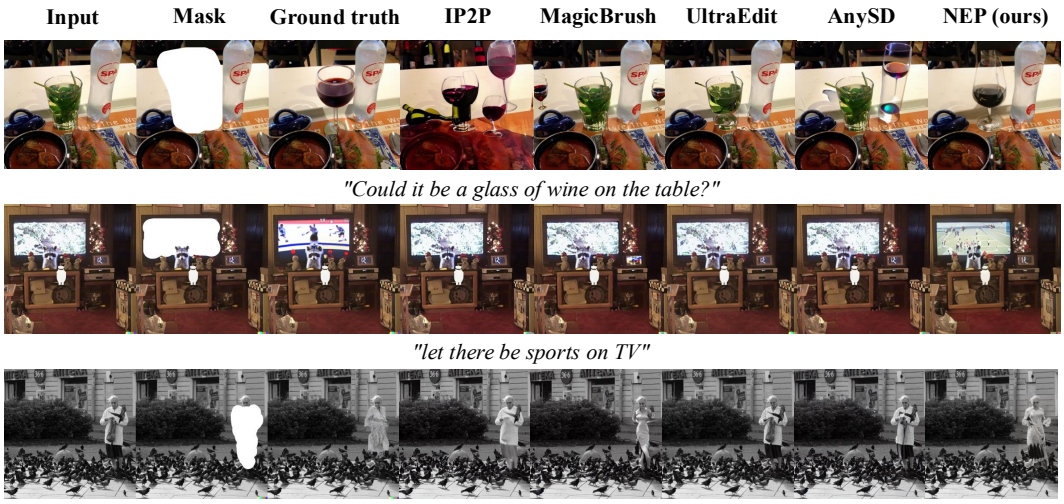

Figure 4: **Comparative editing results.** This demonstrates that our approach can make more faithful edits to source images, either by updating objects (case #1, #2), or making fine-grained edits (case #3). Best viewed zoomed in and in color.

3.3 Results on NEP Pre-training

To better understand how NEP works, we also evaluate the intermediate text-to-image model RLlamaGen obtained during NEP pretraining. RLlamaGen acquires the zero-shot editing ability without sacrificing text-to-image generation performance.

3.3.1 Zero-shot Image Editing

We demonstrate that RLlmaGen is readily capable of image editing. This is achieved by regenerating tokens in the editing regions. Figure 5 demonstrates that RLlmaGen can make fine-grained and coherent edits.

Comparison with Localized Editing Approaches. We compare our zero-shot editing performance against aMUSEd [27], which is also capable of localized zero-shot editing. We use its publicly available checkpoint for comparison, adhering to its default configu-

Table 5: **Comparative Results on Zero-shot Editing on MagicBrush test set.**

Settings	Methods	L1↓	L2↓	CLIP-I↑	DINO↑
Single-turn	aMUSEd	0.0913	0.0300	0.8802	0.8131
	Ours (zero-shot)	**0.0743**	**0.0211**	**0.9032**	**0.8509**
Multi-turn	aMUSEd	0.1034	0.0361	0.8689	**0.8092**
	Ours (zero-shot)	**0.0916**	**0.0319**	**0.8798**	0.7859

https://doi.org/10.52202/085713-4358

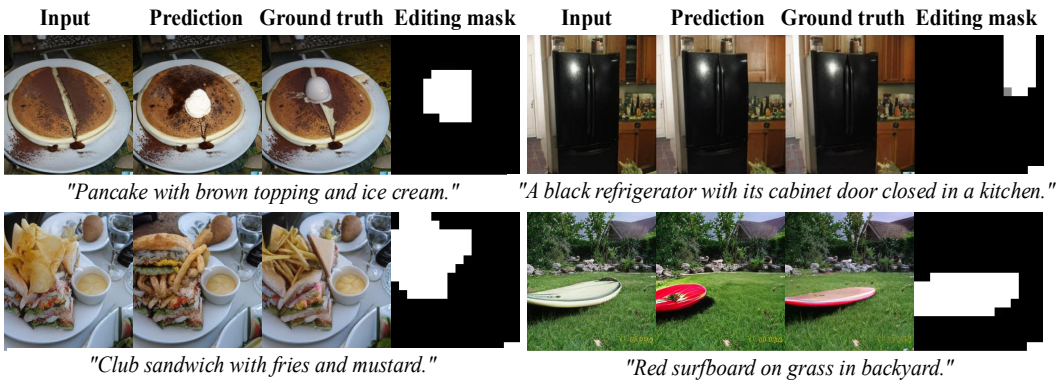

Figure 5: **Examples of RLlamaGen's zero-shot editing capability.** It can make fine-grained edits such as adding external objects (ice cream in example #1), changing the state of input objects (cabinet door open $\rightarrow$ closed in example #2), changing the semantics (chips $\rightarrow$ fries in example #3), and changing the color (white $\rightarrow$ red in example #4). Best viewed zoomed in and in color.

rations[1]. Results on the MagicBrush dataset show that our approach outperforms aMUSEd. This is attributed to our method's ability to enable fine-grained editing by keeping all source image tokens visible to the generation model, whereas aMUSEd replaces edited regions with mask tokens, limiting its precision.

Ablations on Generation Order. Alternative to the default generation order, i.e., an in-mask raster scan order, as we introduced in Section 3.3, we employ random generation order for zero-shot image editing. The results in Table 6 demonstrate that altering the generation order has negligible impact on the effectiveness of our approach, confirming its robustness.

Table 6: **Ablations on Generation Order for Zero-shot Editing on MagicBrush test set.**

Settings	Methods	L1↓	L2↓	CLIP-I↑	DINO↑
Single-turn	In-mask random order	**0.0741**	**0.0211**	0.9027	0.8482
	In-mask raster scan order	0.0743	**0.0211**	**0.9032**	**0.8509**
Multi-turn	In-mask random order	**0.0911**	**0.0316**	0.8782	0.7833
	In-mask raster scan order	0.0916	0.0319	**0.8798**	**0.7859**

3.3.2 Text-to-Image Generation Results

Benchmarks & Evaluation Metrics. We evaluate the image generation quality on MS-COCO 30K in terms of Fréchet Inception Distance (FID) and CLIP similarity. FID reflects the fidelity and diversity of generated images. It measures the distance between the ground truth image distribution and the generated image distribution, where the distributions are constituted of Inception V3 [43] embeddings extracted from corresponding images. The CLIP score is used to evaluate the instruction-following ability of T2I models. It measures the similarity between the vision embeddings extracted from the generated image and text encoder embeddings extracted from corresponding captions.

We demonstrate that randomized pre-training preserves raster scan generation capability. Moreover, employing NEP test-time scaling further improves generation performance. Table 7a shows that RLlamaGen outperforms its baseline (line 2 vs. line 1), and performs similarly with LlamaGen tuned for the same number of steps (line 2 vs. line 3). Scaling NEP for self-refinement can obtain 1.5% improvement in terms of CLIP and 11.4% reduction in FID (line 4 vs. line 2).

3.4 Results on Test-time Scaling of NEP

We evaluate our self-improvement strategy on top of NEP, which iteratively revises the model's previous generation. This self-improvement can be effectively scaled through multi-round iterative refinement. Empirical evidence suggests that masking out previously generated tokens during the revision process yields superior results; thus, we adopt this approach as our default method.

We demonstrate the scaling effects of NEP in Table 7b, where we observe consistent improvements as the number of revision rounds increases. This strategy can be further enhanced by utilizing stronger

[1]https://huggingface.co/blog/amused

https://doi.org/10.52202/085713-4358

Figure 6: **Self-improving RLlamaGen.** By gradually revising the original output, we can obtain images better aligned with instructions and with higher fidelity. Best viewed zoomed in and in color.

Table 7: **Results on NEP pre-training and TTC.** The pre-trained RLlamaGen enables arbitrary order generation without sacrificing generation quality. NEP can be employed for test-time scaling which enhances the generation further.

(a) **Pretraining schemes**. Comparative results between LlamaGen baseline, RLlamaGen fine-tuned for a pre-defined number of steps, and LlamaGen fine-tuned for the same number of steps.

Methods	CLIP↑	FID↓
LlamaGen	0.320	15.07
LlamaGen ft.	0.326	12.00
RLlamaGen	0.325	11.49
TTS w/ NEP	**0.330**	**10.18**

(b) **Test-time scaling w/o post-training.** NEP can be used to iteratively revise generated images. The generation quality gradually improves and saturates after 2 iterations.

# Revision rounds	CLIP↑	FID↓
0	0.325	11.49
1	**0.332**	9.94
2	**0.332**	9.93
3	**0.332**	9.85
4	**0.332**	**9.82**

verifier models and training the model for self-improvement. The revision process is visualized in Figure 6, showcasing better alignment with the conditioning text prompts and higher fidelity.

4 Related Works

4.1 Text-to-Image Generation

Text-to-image generation has become a cornerstone of modern artificial intelligence, enabling to create visual content based on textual descriptions. Pioneering models such as Generative Adversarial Networks (GANs) [10] make groundbreaking breakthroughs by generating high-fidelity images. AttnGAN [52] built on StackGAN [57] achieves better alignment with text instructions. However, GANs still faced challenges like training instability (e.g., mode collapse, where the model generates limited varieties of images) and difficulty with highly detailed or multi-object scenes, setting the stage for the next evolutionary step.

More recently, diffusion models [37, 13, 38] like Stable Diffusion [32] have emerged, creating realistic images by iteratively denoising random noise guided by text descriptions, setting a new standard for quality and versatility. However, the learning paradim and architectures diverge from well-established large language models (LLMs) [3],, making it difficult for artificial general intelligence featuring a shared framework for various modalities.

In this regard, a line of works [28, 29, 53] resort to autoregressive models for visual generation. Images are tokenized into a sequence of tokens and generated sequentially based on prefilled text tokens. Benefiting from large-scale models and datasets, they can create photorealistic images with a remarkable text-following capability. This field is further advanced by several open-source works, such as LlamaGen [42], Emu3 [48], and Janus [49].

https://doi.org/10.52202/085713-4358

4.2 Image Editing

Image editing builds on text-to-image generative models by conditioning outputs on source images, but preserving unedited regions poses a challenge for diffusion models. These models require looking for mapping latent representations for the original RGB values, often using inversion techniques [39, 21]. However, such methods typically demand inference-time tuning, such as tuning textual embeddings [9], model weights [33, 47], or null-text embeddings [21] to enable classifier-free guidance [12]. Even when noise trajectories across varying levels are available, maintaining unedited regions is not assured. For instance, Prompt-to-Prompt [11] introduces a time threshold to prioritize generating target object geometry through text-to-image steps without source image conditioning, trading off reconstruction accuracy for generative flexibility.

Efforts to guide edits using user-specified masks have been explored in both training-free [1] and training-based approaches [23, 58, 60]. Training-free methods apply masks across all diffusion steps to blend source image latents with text-conditioned outputs, while training-based methods append an extra channel to the source image for guidance. Despite these advancements, both approaches require full image regeneration, which hampers efficiency during training and inference.

In contrast, our work enables localized editing by regenerating tokens solely within user-defined regions, preserving pixels outside these areas without modification. Leveraging user-provided masks introduces minimal limitations, thanks to recent advances in segmentation techniques [15, 30, 16].

4.3 Test-time Scaling for Text-to-Image Generation

The success of LLMs' inference-time scaling motivates the exploration of similar behavior for text-to-image generation. Existing approaches mainly investigate diffusion model scaling, either by increasing the denoising step [14, 41] or employing best-of-N sampling [19]. More recently, new test-time scaling approaches have emerged that enable revising prior generations by incorporating corrections and feedback into the context [17]. However, an additional post-training stage is required to support their iterative refinement, limiting their flexibility and increasing computational demands. In this work, we investigate inference-time scaling in autoregressive image generation models that can conduct self-improvement utilizing NEP, offering a new perspective on enhancing model performance during testing without dedicated post-training.

5 Conclusion

In this work, we propose a next-editing token-prediction pipeline for text-driven image editing. It allows for easy localized editing without making unintended modifications to the non-editing region. To support regeneration at any user-specified position, we pre-train an any-order autoregressive T2I model that can generate tokens in arbitrary orders. Furthermore, we demonstrate NEP can be integrated into an iterative refinement loop for test-time scaling.

6 Limitations and Broader Impacts

Limitations. While the proposed approach demonstrates promising results, it relies on user-provided masks for guidance to prevent unintended modifications to the source image. This requirement adds extra computation or annotation, making the process less efficient. We plan to address automated and unified masking region localization in future work. Additionally, the robustness of Neural Editing Propagation (NEP) to noise in editing region masks remains uncertain. Imperfect user-specified masks lead to two primary scenarios: 1) the segmentation mask is larger than the ground truth editing region, and 2) the segmentation mask is smaller. In the first scenario, NEP exhibits robustness, achieving comparable results, as shown in Table 2 for free-form image editing without a mask. In the second scenario, our approach lacks specific optimization. We plan to develop a pipeline for automatically refining user-specified masks in future work.

Social impacts. Our primary motivation for developing image editing algorithms is to foster innovation and creativity; however, we recognize that they also present significant ethical and societal challenges. We are committed to minimizing these risks by filtering training images for unsafe content and restricting the model's use to research purposes only upon release. In the future, we will actively engage in discussions and initiatives aimed at mitigating these risks.

References

[1] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 18187–18197. IEEE, 2022. URL https://doi.org/10.1109/CVPR52688.2022.01767.

[2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

[4] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021.

[5] Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. Allava: Harnessing gpt4v-synthesized data for lite vision-language models. *arXiv preprint arXiv:2402.11684*, 2024.

[6] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pages 370–387. Springer, 2024.

[7] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[8] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 12873–12883. Computer Vision Foundation / IEEE, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Esser_Taming_Transformers_for_High-Resolution_Image_Synthesis_CVPR_2021_paper.html.

[9] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144, 2020.

[11] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *CoRR*, abs/2208.01626, 2022. URL https://doi.org/10.48550/arXiv.2208.01626.

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

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[14] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

[15] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

 https://doi.org/10.52202/085713-4358

[16] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2024.

[17] 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.

[18] Guangyi Liu. laion-coco-aesthetic. `https://huggingface.co/datasets/guangyil/laion-coco-aesthetic`, 2023.

[19] Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

[20] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=aBsCjcPu_tE`.

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

[22] Jiteng Mu, Nuno Vasconcelos, and Xiaolong Wang. Editar: Unified conditional generation with autoregressive models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7899–7909, 2025.

[23] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 16784–16804. PMLR, 2022. URL `https://proceedings.mlr.press/v162/nichol22a.html`.

[24] Xichen Pan, Li Dong, Shaohan Huang, Zhiliang Peng, Wenhu Chen, and Furu Wei. Kosmos-g: Generating images in context with multimodal large language models. In *ICLR*, 2024.

[25] Ziqi Pang, Tianyuan Zhang, Fujun Luan, Yunze Man, Hao Tan, Kai Zhang, William T Freeman, and Yu-Xiong Wang. Randar: Decoder-only autoregressive visual generation in random orders. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 45–55, 2025.

[26] Arnaud Pannatier, Evann Courdier, and François Fleuret. σ-gpts: A new approach to autoregressive models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 143–159. Springer, 2024.

[27] Suraj Patil, William Berman, Robin Rombach, and Patrick von Platen. amused: An open muse reproduction. *arXiv preprint arXiv:2401.01808*, 2024.

[28] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR, 2021. URL `http://proceedings.mlr.press/v139/ramesh21a.html`.

[29] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022. URL `https://doi.org/10.48550/arXiv.2204.06125`.

[30] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[33] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22500–22510, June 2023.

[34] Christoph Schuhmann and Peter Bevan. 220k-gpt4vision-captions-from-lvis. `https://huggingface.co/datasets/laion/220k-GPT4Vision-captions-from-LIVIS`, 2023.

[35] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[36] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. *arXiv preprint arXiv:2311.10089*, 2023.

[37] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

[38] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[39] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=St1giarCHLP`.

[40] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022.

[41] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[42] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

[43] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[44] Xueyun Tian, Wei Li, Bingbing Xu, Yige Yuan, Yuanzhuo Wang, and Huawei Shen. Mige: A unified framework for multimodal instruction-based image generation and editing. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025.

[45] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[46] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[47] Dani Valevski, Matan Kalman, Yossi Matias, and Yaniv Leviathan. Unitune: Text-driven image editing by fine tuning an image generation model on a single image. *arXiv preprint arXiv:2210.09477*, 2(3):5, 2022.

[48] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

[49] 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. *arXiv preprint arXiv:2410.13848*, 2024.

[50] 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. *arXiv preprint arXiv:2409.11340*, 2024.

 https://doi.org/10.52202/085713-4358

[51] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

[52] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018.

[53] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022.

[54] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[55] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. *arXiv preprint arXiv:2411.00776*, 2024.

[56] Chenhui Zhang. laion-coco-17m. `https://huggingface.co/datasets/danielz01/laion-coco-17m`, 2024.

[57] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017.

[58] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36, 2024.

[59] Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, et al. Hive: Harnessing human feedback for instructional visual editing. *arXiv preprint arXiv:2303.09618*, 2023.

[60] 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.

[61] Jun Zhou, Jiahao Li, Zunnan Xu, Hanhui Li, Yiji Cheng, Fa-Ting Hong, Qin Lin, Qinglin Lu, and Xiaodan Liang. Fireedit: Fine-grained instruction-based image editing via region-aware vision language model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We make sure that the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Section 6 for the discussion of this work's limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

https://doi.org/10.52202/085713-4358

Justification: Please note that this work does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: This paper contains high-level demonstration of our approach in Fig. 2 and its detailed descriptions in Section 2. Implementational details are released in Section 3 including training dataset, configurations, hyperparameters, and evaluations. The code will be released upon acceptance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All datasets used in this work are open-sourced, and we will release our code upon acceptance with detailed instructions so that this work can be faithfully reproduced.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The training details of T2I and image editing are elaborated in Section 3.1, including the dataset, network architectures, training steps, batch size, optimizer, and learning rate. Their evaluation details are presented in Section 3.2 and 3.3.2, respectively.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Following common practices in image editing/generation research, and for a fair comparison, we did not include statistical significance tests.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

https://doi.org/10.52202/085713-4358

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient information on the computing resources required to reproduce our experiments in Section 3.1. This section includes details about the types of GPU cards used, the number of them, and the training time for each experiment.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We fully adhere to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the social impacts of this work in Section 6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We discuss these measurements in Section 6, mainly by filtering training images to keep only safe content. When we release the code, the model will be licensed for research purposes only, minimizing the risk of misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have made every effort to ensure that all creators and original owners of the assets used in our paper are properly credited, and we have respected their licenses and terms of use throughout our work.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

https://doi.org/10.52202/085713-4358

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The main asset contributed by this work is the source code. It will be released after this paper is accepted with detailed documentation.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

https://doi.org/10.52202/085713-4358