

# EditAR: Unified Conditional Generation with Autoregressive Models

Jiteng Mu<sup>1</sup>, Nuno Vasconcelos<sup>1</sup>, Xiaolong Wang<sup>1,2</sup>

<sup>1</sup>UC San Diego, <sup>2</sup>NVIDIA

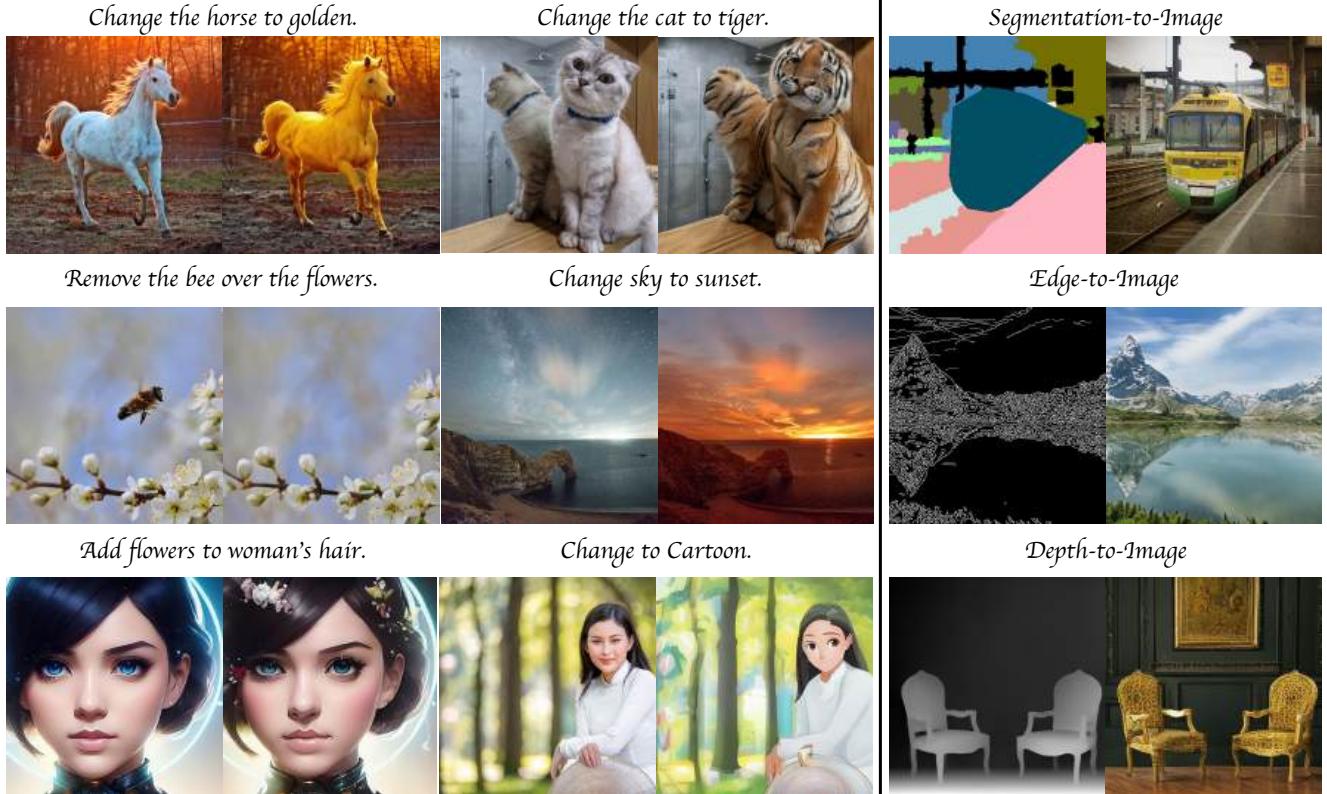


Figure 1. We propose EditAR, a unified conditional autoregressive model for diverse conditional generation tasks. We demonstrate that without task-specific designs, a single autoregressive model achieves strong performance across diverse tasks, including texture manipulation, object replacement, object removal, local editing, canny-to-image, depth-to-image, and segmentation-to-image.

## Abstract

Recent progress in controllable image generation and editing is largely driven by diffusion-based methods. Although diffusion models perform exceptionally well in specific tasks with tailored designs, establishing a unified model is still challenging. In contrast, autoregressive models inherently feature a unified tokenized representation, which simplifies the creation of a single foundational model for various tasks. In this work, we propose **EditAR**, a single unified autoregressive framework for a variety of conditional image generation tasks, e.g., image editing, depth-to-image, edge-to-image, segmentation-to-image. The model takes both images and instructions as inputs, and predicts the edited images tokens in a vanilla next-token paradigm. To en-

hance the text-to-image alignment, we further propose to distill the knowledge from foundation models into the autoregressive modeling process. We evaluate its effectiveness across diverse tasks on established benchmarks, showing competitive performance to various state-of-the-art task-specific methods. Project page: <https://jitengmu.github.io/EditAR/>

## 1. Introduction

Recent advances in conditional image generation are predominantly driven by high capacity text-to-image diffusion models [24, 25, 41, 48, 52, 55, 57, 58]. These models serve as a strong image prior, that can be specialized to individual tasks, facilitating remarkable progress in applications

like inpainting [35, 69, 70], personalization [18, 29, 56], image editing [4, 22, 28, 38, 40], and image translation [30, 74]. While these approaches excel at each individual task, the resulting variations in architectural designs and learning objectives make it challenging to integrate multiple tasks within a single framework. Ideally, a single conditional model would excel across all tasks, simplifying both implementation and deployment.

Large-scale autoregressive models have recently gained attention as a promising path towards this goal [61, 63, 71, 72]. Unlike diffusion models, they naturally provide a unified, token-based, framework to blend diverse inputs. This unified representation offers substantial promise for creating a single model capable of performing a wide range of image synthesis and manipulation tasks within a unified architecture. While recent works [61, 63] demonstrate remarkable text-to-image generation performance, it is still unclear whether such models are well-suited as foundational architectures for broader conditional generation tasks. While a few autoregressive variants [10] have demonstrated the possibility of image inpainting by modifying the next-token prediction paradigm, the efficacy of next-token prediction has not yet been demonstrated for image editing.

In this work, we investigate the feasibility of building a single autoregressive model that unifies various conditional image generation tasks. We introduce **EditAR**, a novel autoregressive model that, for the first time, integrates various image manipulation and image translation tasks. Like prior text-to-image autoregressive models [10, 61, 63], it consists of two stages: a VQVAE [54, 67], that maps image patches into tokens indices, and an autoregressive transformer [15] that models the categorical distribution of output tokens given both texts and images as inputs. As illustrated in Figure 1, we show that that this achieves promising performance *across a significant variety of tasks*.

EditAR builds primarily on Llamanen [61], a text-to-image autoregressive model based on the Llama2 [64, 65] architecture that has demonstrated impressive image generation capabilities. However, due to the lack of a conditional image input, Llamanen does not support tasks like image manipulation or translation. To allow this, we adapt the architecture by prefilling the model with image tokens from a conditioning input image, along with additional positional embeddings. This is complemented by an auxiliary distillation loss, based on DINOv2 [9, 44], which reinforces the visual coherence of the images synthesized by the autoregressive model. Empirical results show that injecting this visual prior enhances alignment between the generated images and the input text.

At inference, given an image and corresponding editing instructions, output tokens are generated by the standard next-token prediction paradigm. To enhance image quality and text-image alignment, classifier-free guidance [23]

is applied to both image and texts. To our knowledge, this paper provides the first evidence for the feasibility of using next-token prediction autoregressive models for conditional generation on large benchmarks. Furthermore, by solving diverse conditional image generation tasks, it paves the way for a new class of approaches to unified conditional generation. Our contributions are as follows:

- We introduce a new autoregressive framework EditAR, that is jointly trained on various image manipulation and image translation tasks, and demonstrates promising potential towards building a unified conditional image generation model.
- A distillation loss is introduced to enhance the semantics in the learning of autoregressive models.
- Experiments show that the proposed method demonstrates strong performance on a variety of tasks, including texture manipulation, object replacement, object removal, local editing, canny-to-image, depth-to-image, and segmentation-to-image.

## 2. Related Work

### 2.1. Controllable Image Diffusion

Text-to-image diffusion models [2, 11, 14, 16, 23–25, 34, 41, 48, 49, 53, 55, 57–60] have revolutionized image generation, significantly improving the realism of the generated visuals. Nevertheless, adapting these models for tasks like image editing poses notable challenges, as synthesized images can include artifacts, fail to follow the editing instruction, or fail to remain faithful to the conditioning image.

One popular approach [22, 28, 36–38, 47, 68] is to frame editing as a two stage process: the conditioning image is first inverted into noise, which is then run through the forward diffusion chain conditioned by the editing instructions. For example, SDEdit [36] adds noise to the image, which is then progressively denoised via a stochastic differential equation (SDE). This, however, can struggle to reconstruct the image details that are supposed to remain unaltered. More advanced techniques improve both the inversion and content-preserving generation quality. To improve the reconstruction quality, optimization-based inversion methods [28, 37, 38, 68] are proposed to invert the conditioning image into a latent embedding that achieves almost perfect reconstruction. This is then coupled with content-preserving generation techniques, usually based on the preservation of cross-attention maps with text prompts [8, 22, 47]. While producing higher-quality visuals, these methods require additional computation and time for the optimization of the latent embeddings.

A faster alternative is to take a strictly feedforward approach [4, 40, 73], directly incorporating both conditioning image and texts in the denoising process of the diffusion model. For example, InstructPix2Pix [4] generates large-

scale paired data of conditioning, edited image, and editing instructions, using GPT-3 [5] and Prompt-to-Prompt [22]. These data are used to train a text-to-image diffusion model in a supervised way, demonstrating strong capabilities for texture transfer and object replacement. However, this type of approach is not suitable for translation tasks involving sparse conditioning signals, such as canny or edge maps. Better performance for these tasks can be achieved by approaches [31, 39, 50, 74, 75] that, like ControlNet, introduce additional trainable parameters to create a translation model. However, these parameters usually need to be trained for each type of conditioning modality, and multiple models are needed to support multiple modalities. UniControlNet [75] and UniControl [50] advanced this by unifying various sparse conditions in a single model and shows promising potentials. Differently, our method jointly solve both the image editing and translation task, significantly simplifying the design and deployment of real-world applications.

## 2.2. Autoregressive Models

Autoregressive models are a class of generative models that treat text-to-image generation as a sequence-to-sequence modeling problem, similar to machine translation. Early works like VQVAE [67], pioneered the idea of using vector quantization for this purpose, allowing an image to be encoded as a sequence of discrete tokens, as is common for language. This enables the model to process visual data similarly to language tokens, and paved the way for a series of transformer-based autoregressive models [10, 15, 52, 61, 62, 72]. Recently, with the rise of large language models (LLMs) [1, 6, 12, 13, 20, 32, 33, 42, 43, 45, 64, 65], LLM-style autoregressive models have started to gain popularity for text-to-image generation. Approaches like Parti [52, 62, 72] leverage large scale data and model sizes to successfully synthesize high quality, visually diverse images. Yet, most of these approaches remain focused on text-to-image generation, with limited efforts in adapting these models for conditional generation tasks.

Other works [10, 63, 76] improve autoregressive models by reconsidering the next-token prediction paradigm. They challenge the line-by-line raster-scan generation of image tokens, arguing that images require more global contexts than text. Inspired by Masked autoencoders [21], MaskGIT [10] uses an iterative masked modeling approach. It learns to predict randomly masked tokens by attending to tokens and iteratively decodes tokens during inference. VAR [63] changes the next-token prediction paradigm into a coarse-to-fine next-scale prediction, significantly improving the visual quality of the generated images.

The model proposed in this work builds on LlamaGen [61], which generalizes the Llama2 [64, 65] architecture and is currently the state-of-the-art open-source autoregressive model for text-to-image generation. The proposed model retains its next-token prediction paradigm, leveraging LlamaGen as a strong prior for text-to-image generation, and further augments it, by incorporating both images and texts as inputs and extending its capabilities to a much wider range of conditional generation tasks.

In image editing, there has been a recent trend to combine LLMs with diffusion models [3, 17, 26, 76], thus addressing the limitations of CLIP-type text encoders widely used in text-to-image diffusion. These works have argued [17, 26] that replacing the CLIP [51] text encoder with LLM modules [32, 33] enhances the reasoning capabilities of the diffusion model, enabling more nuanced interpretation of the editing instructions. However, the combination of the two models is memory, computation, and time intensive. Our aim is to design a pure autoregressive model that directly outputs the edited image as a sequence of tokens, rather than outputs in LLM text output space, thus removing the need for diffusion models. This has much lower complexity and avoids the challenges of jointly optimizing models that are trained in fundamentally different ways (autoregressive for text, denoising for images). Combined models have also only been shown successful for editing, while the proposed model also supports translation tasks involving different types of conditioning signals, e.g. the sparse signals of edge-to-image translation. This makes the proposed architecture a unified solution to a much broader range of tasks.

## 2.3. Combining LLM with Diffusion Models

In image editing, there has been a recent trend to combine LLMs with diffusion models [3, 17, 26, 76], thus addressing the limitations of CLIP-type text encoders widely used in text-to-image diffusion. These works have argued [17, 26] that replacing the CLIP [51] text encoder with LLM modules [32, 33] enhances the reasoning capabilities of the diffusion model, enabling more nuanced interpretation of the editing instructions. However, the combination of the two models is memory, computation, and time intensive. Our aim is to design a pure autoregressive model that directly outputs the edited image as a sequence of tokens, rather than outputs in LLM text output space, thus removing the need for diffusion models. This has much lower complexity and avoids the challenges of jointly optimizing models that are trained in fundamentally different ways (autoregressive for text, denoising for images). Combined models have also only been shown successful for editing, while the proposed model also supports translation tasks involving different types of conditioning signals, e.g. the sparse signals of edge-to-image translation. This makes the proposed architecture a unified solution to a much broader range of tasks.

## 3. Method

This work aims to create an autoregressive model that unifies the solution for various conditional image synthesis tasks, including different types of image editing, and image translation tasks such as edge-to-image, or segmentation-to-image. For this, we leverage existing large-scale text-to-image autoregressive models [61], which we generalize to integrate both image and text elements as conditions. Additionally, a training strategy is proposed to effectively unify various conditions, enabling high-quality image generation. In what follows, we first briefly review the paradigm of text-to-image autoregressive models in Section 3.1, which forms the basis of our approach. We then describe modifications, in Section 3.2, to incorporate image conditioning beyond texts. Finally, we present effective learning and inference strategies for the proposed model in Section 3.3.

### 3.1. Background

Autoregressive models [15] approach text-to-image generation as a sequence-to-sequence modeling task. A common approach includes two main components: a VQ-Autoencoder [54, 67] that converts images into discrete tokens, and an autoregressive transformer [15] that models the

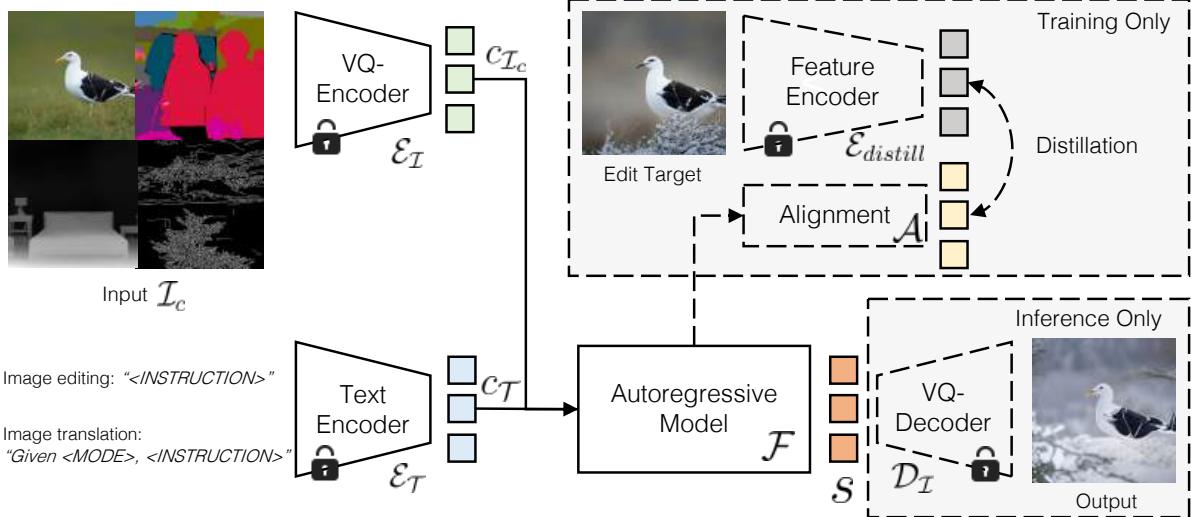


Figure 2. Overview of EditAR, which can take various types of image conditions to perform image editing or translation. An image  $\mathcal{I}_c$  is mapped through a VQ-Encoder  $\mathcal{E}_I$  to obtain corresponding token indices. Corresponding text instructions are mapped to latent embeddings  $c_T$  via a text encoder  $\mathcal{E}_T$ . Both image token indices and text embeddings are input to the autoregressive transformer  $\mathcal{F}$  to predict the target token indices  $s$ . To enhance the text-to-image alignment, a distillation loss is introduced during training to minimize the differences between the latent features of the autoregressive model,  $\mathcal{F}$  and that of a feature encoder  $\mathcal{E}_{distill}$ . The output sequence  $s$  is lastly decoded into a realistic image via a VQ-Decoder  $\mathcal{D}_I$  during inference.

categorical distribution of these tokens.

A VQ-Autoencoder uses an encoder  $\mathcal{E}_I$  to map an image  $\mathcal{I} \in \mathcal{R}^{H \times W \times 3}$  into a latent feature map  $z \in \mathcal{R}^{h \times w \times n_z}$  of feature dimensionality  $n_z$ . A vector quantizer is then used to map each feature vector  $z_{i,j} \in \mathcal{R}^{n_z}$  into its nearest neighbor  $z_{i,j}^q$  in a feature vector codebook. This allows the representation of the image as a sequence of discrete codebook indices,  $s = \{s_1, s_2, \dots, s_{h \cdot w}\}$ . A decoder  $\mathcal{D}_I$  can finally be used to map these indices into the corresponding codebook entries, to recover the image.

One of the advantages of this quantization operation is that the index sequence  $s$  is not fundamentally different from the sequences of one-hot codes commonly used to represent sentences in natural language. This allows a natural unified treatment of the two modalities. While large language models are autoregressive models that map a text token sequence into another text token sequence, a text-to-image generation autoregressive model outputs a sequence of visual tokens. Hence, the task of a text-to-image generation autoregressive model  $\mathcal{F}$  reduces to modeling the distribution of the next index  $p(s_i | s_{<i}, c_T)$ , where  $c_T$  denotes the text embeddings obtained from a text encoder  $\mathcal{D}_T$ . The likelihood of the full sequence is thus defined as,

$$p(s) = \prod_{i=1}^n p(s_i | s_{<i}, c_T), \quad (1)$$

and the model parameters can be learned by maximizing the log-likelihood of the token data,  $\mathbb{E}_{s \sim p(s)}[-\log p(s)]$ .

### 3.2. EditAR

The proposed architecture for conditional image synthesis builds on LlamaGen [61], which is generalized to incorporate various types of image conditioning and a distillation loss for alignment with existing foundation models. The overall architecture is illustrated in Figure 2 and has the key design choices outlined below.

**Image Condition.** The conditioning image  $\mathcal{I}_c$  is mapped to a sequence of indices,  $c_{\mathcal{I}_c} = \{c_1, c_2, \dots, c_{h \cdot w}\}$ , via an image encoder  $\mathcal{E}_I$ . The same encoder  $\mathcal{E}_I$  is also used to map the target image  $\mathcal{I}_t$  into index sequence  $s$ . The likelihood of the output sequence  $s$  becomes

$$p(s) = \prod_{i=1}^n p(s_i | s_{<i}, c_T, c_{\mathcal{I}_c}). \quad (2)$$

This is implemented with the extension of a text-to-image autoregressive model, as shown in Figure 2. Both the additional image indices  $c_{\mathcal{I}_c}$  and the text embeddings  $c_T$  are fed into the autoregressive model  $\mathcal{F}$ . During training, they are complemented by the indices  $s$  that appear at both input  $s_{<i}$  and output  $s_i$  of the model to implement the autoregressive operation. Note that different sets of positional embeddings are applied to the embeddings of  $c_{\mathcal{I}_c}$  and  $s$  to differentiate the control image sequence from the output sequence. In addition, an unconditional image embedding is introduced and used during training to preserve the model's text-to-image and unconditional generation ability.

**Incorporating Image Modalities.** Our method supports various imaging modalities, including canny edges, depth

maps, segmentation masks, and natural images. The generation process is adjusted to these different modalities by modifying the phrasing of text inputs. For instance, to generate an image from a depth map, we use the “*Given the depth, generate the image following the instruction: <INSTRUCTION>*”, where the instruction is usually the description of the content of the generated image. This approach is similarly applied for canny edge maps and segmentation masks. When the input is a real image, we use only “<INSTRUCTION>” to specify how to modify the input image.

**Distillation.** Distillation from vision foundation models has been demonstrated to be effective across a variety of computer vision tasks. An autoregressive model gradually learns to synthesize tokens of high likelihood given the conditioning token stream. However, the autoregressive model, trained solely to predict token indices, are not guaranteed to learn general semantic features. To inject general visual knowledge in the feature space, we introduce a distillation loss from the vision foundation model feature encoder  $\mathcal{E}_{distill}$ , which in our implementation is DINOv2 [9, 44]. An alignment network  $\mathcal{A}$ , composed of a single convolutional layer, is used to match the dimensionality of the embedding space of the autoregressive model  $\mathcal{F}(\cdot)$  with that of the foundation model. During training, the parameters of this network are learned to minimize the distillation loss

$$\mathcal{L}_{distill} = MSE\left(\mathcal{A}(\mathcal{F}(\cdot)), \mathcal{E}_{distill}(\cdot)\right). \quad (3)$$

For both  $\mathcal{F}$  and  $\mathcal{E}_{distill}$ , the features extracted from the last hidden layer are used to compute this loss. Empirically, we find this design to improve the text-to-image alignment.

### 3.3. Training and Inference

During training, both the text-encoder  $\mathcal{E}_T$  and foundation model  $\mathcal{E}_{distill}$  are frozen. The parameters of the autoregressive transformer  $\mathcal{F}$  are initialized from pre-trained text-to-image models. All parameters of  $\mathcal{F}$  and  $\mathcal{A}$  are optimized for the adaptation, using

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda_{distill} \cdot \mathcal{L}_{distill}, \quad (4)$$

where  $\mathcal{L}_{CE} = \mathbb{E}_{x \sim p(x)}[-\log p(s)]$  is the cross-entropy loss commonly used to train next-token prediction models.

Motivated by prior works [4, 61], we apply dropout during training to preserve the model’s unconditional generation ability, and classifier-free guidance during inference. For this, we set  $c_T = \emptyset$  in 5% of the training examples,  $c_{I_c} = \emptyset$  in another 5%, and both  $c_T = \emptyset$  and  $c_{I_c} = \emptyset$  in a final 5%. At inference, only  $c_{I_c}$  and  $c_T$  are given as inputs, and the set  $s$  is predicted sequentially. Classifier-free guidance consist of using

$$\begin{aligned} \log p(s_i | s_{<i}, c_T, c_I) &= \log p(s_i | s_{<i}, c_I) \\ &+ \eta \cdot \left( \log p(s_i | s_{<i}, c_T, c_I) - \log p(s_i | s_{<i}, c_I) \right), \end{aligned} \quad (5)$$

where  $\eta$  is a guidance strength hyperparameter, to compute index log-probabilities at inference.

## 4. Experiments

In this section, we discuss an extensive experimental evaluation of EditAR on various conditional image synthesis tasks. Note that EditAR is a single model that solves both image editing, as discussed in Section 4.2, and image translation, as discussed in Section 4.3, while all baseline models we compare to are specialized models for a subset of tasks. Surprisingly, despite this disadvantage, EditAR still shows strong performance.

### 4.1. Dataset and Training Details

**Dataset.** EditAR is trained in a fully supervised manner using paired data. Recent works [4, 19, 27] have shown that it is possible to create large-scale image editing data with automated pipelines. We use 1.5M examples from SEED-Data-Edit-Unsplash [19], created with ChatGPT [42] and Plug-and-play [66], for a range of image editing tasks, including modifying styles, objects, colors, and materials. To further support editing operations like object addition and removal, we add the PIPE dataset [69] with 1.8M examples, where image pairs for the task of adding objects are created by inpainting. During training, we randomly flip each pair with a 50% probability and adjust the editing instruction from “Add” to “Remove” correspondingly. For image translation tasks, we follow ControlNet++ [30] and use COCOSTuff [7] for segmentation mask-to-image translation and MultiGen-20M for canny edge and depth-to-image tasks.

**Evaluation and Metrics.** To systematically evaluate EditAR, we use multiple benchmark datasets. For image editing, we use the PIE-Bench dataset [28] with 700 examples, covering 10 editing types. Our method uses the source image and editing instructions to predict the target edit. Both reconstruction and text-to-image alignment are evaluated as in [28] with the annotated foreground masks. For the evaluation of image translation, we follow ControlNet++ [30] and use the corresponding validation splits for COCOSTuff [7] and MultiGen-20M [50], which contain 5,000 examples per task. Regarding metrics, we follow the common practice in the field: mIOU is used for semantic segmentation conditions, RMSE for depth map conditions, and SSIM for canny edge conditions. For canny edge condition, (100, 200) are used as thresholds.

**Training and Inference.** The training hyperparameters follow the setting of [61]. All images are resized to a resolution of  $512 \times 512$  for both training and inference. The VQ-Autoencoder has a downsampling ratio of 16, so that each image is represented by 1024 tokens. The model is optimized using AdamW with a constant learning rate of  $10^{-4}$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.95$ , and weight decay of 0.05.

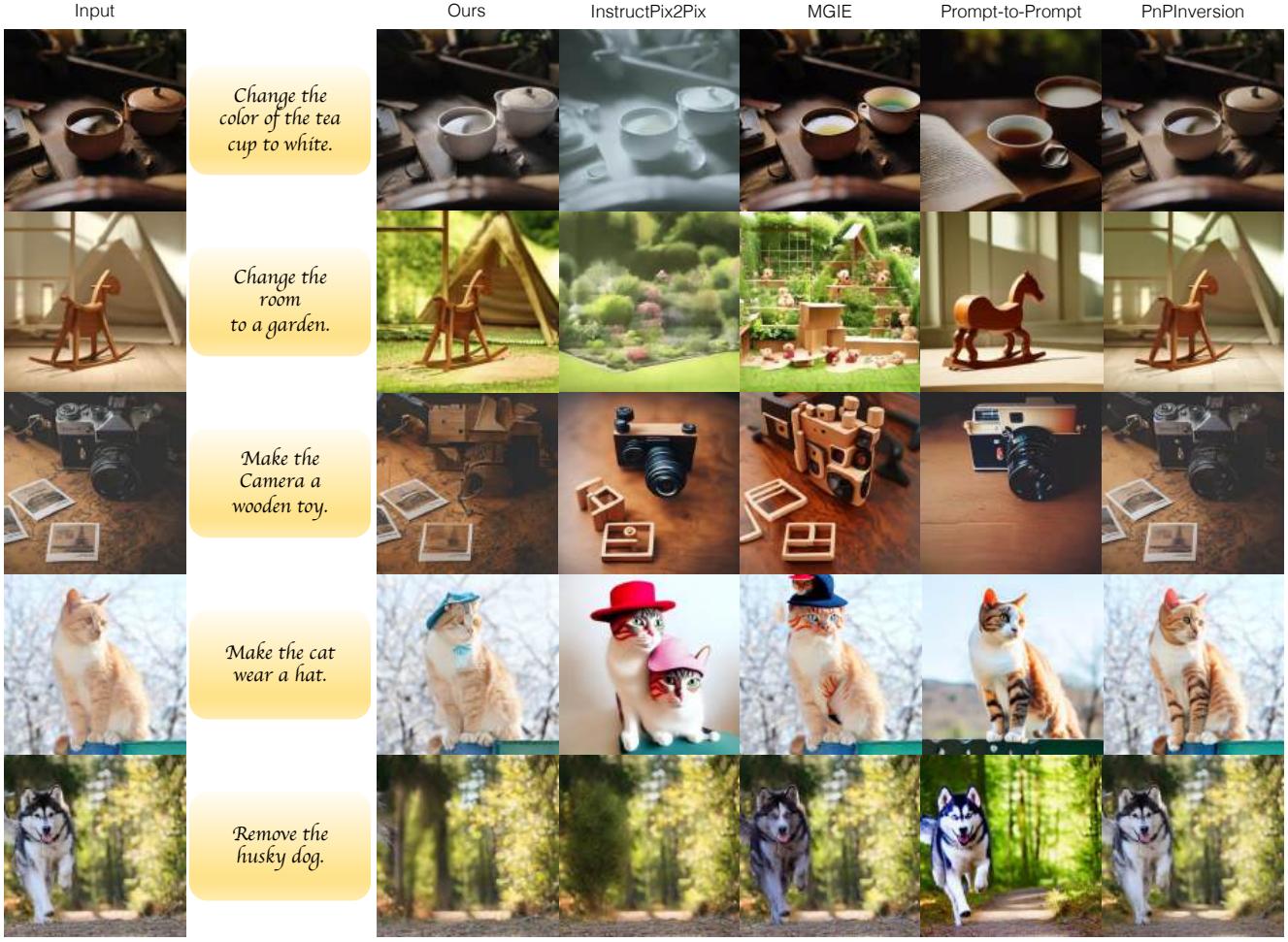


Figure 3. Comparison of EditAR (Ours) to feed-forward methods (InstructPix2Pix [4], MGIE [17]) and inversion-based approaches (Prompt-to-Prompt [22], PnPInversion [28]) on various edits. Our method preserves input details well and has strong text-to-image alignment. In contrast, baseline results exhibit unrealistic visuals, including exaggerated edits or not following instructions, unrealistic modifications, or are unable to localize objects accurately.

The model is trained with a batch size of 64 for 40,000 iterations. In all experiments, we use  $\lambda_{distill} = 0.5$  in (4). For the guidance strength of (5), we empirically find that  $\eta = 3.0$  yields a good balance between reconstruction quality and text-to-image alignment. More details and results are presented in the supplementary materials.

## 4.2. Image Editing

For image editing, we compare EditAR to various baselines, as shown in Table 1 and Figure 3. Specifically, we consider two types of diffusion baselines: feed-forward instruction-based methods and inversion-based content-preserving editing methods.

EditAR achieves the best overall performance among feed-forward instruction-based baselines, e.g., InstructPix2Pix [4] and MGIE [17]. All methods share the same inputs: an input image and an instruction prompt. Instruct-

Pix2Pix excels particularly in texture transfer and object replacement tasks. MGIE leverages both multimodal large language models (MLLMs) to generate expressive instructions, which provides explicit guidance to enhance the editing process. However, as shown in Table 1 and Figure 3, while these methods achieve great text-to-image alignment, they often struggle to preserve the background or produce exaggerated image edits. In comparison, EditAR maintains a good balance between reconstruction and editing quality, resulting in highly photorealistic images.

Unlike feed-forward methods, inversion-based approaches first invert an image into a latent space, usually latent noise or embeddings, before performing content-preserving sampling to generate the edited target. Instead of directly taking an editing instruction, these methods additionally require a source prompt, which is crucial for the inversion process. As shown in Table 1, though Prompt-

Method	T2I Model	Structure	Background Preservation				CLIP Similarity	
		Distance ↓	PSNR ↑	LPIPS ↓	MSE ↓	SSIM ↑	Whole ↑	Edited ↑
Prompt-to-Prompt	SD1.4	69.43	17.87	208.80	219.88	71.14	25.01	22.44
Null-text Inversion	SD1.4	13.44	27.03	60.67	35.86	84.11	24.75	21.86
PnPInversion	SD1.4	11.65	27.22	54.55	32.86	84.76	25.02	22.10
Pix2pix-zero	SD1.4	61.68	20.44	172.22	144.12	74.67	22.80	20.54
MasaCtrl	SD1.4	28.38	22.17	106.62	86.97	79.67	23.96	21.16
InstructPix2Pix	SD1.5	107.43	16.69	271.33	392.22	68.39	23.49	<b>22.20</b>
MGIE	SD1.5	67.41	21.20	142.25	295.11	<b>77.52</b>	24.28	21.79
EditAR (Ours)	LlamaGen	<b>39.43</b>	<b>21.32</b>	<b>117.15</b>	<b>130.27</b>	75.13	<b>24.87</b>	21.87

Table 1. Comparison of EditAR to various feed-forward methods (bottom) and inversion-based approaches (top) on the PIE-Bench dataset [28]. Our method achieves the highest overall performance among all feed-forward approaches and narrows the gap with advanced inversion-based methods. While InstructPix2Pix [4] achieves a high edited CLIP score, it struggles to reconstruct unedited regions accurately, as indicated by its lower whole CLIP score and background scores. MGIE [17] shows improved background preservation and editing quality, yet our method demonstrates stronger overall performance.

Method	T2I Model	Depth Map (MultiGen)		Canny Edge (MultiGen)		Seg. Mask (COCOStuff)	
		RMSE ↓	FID ↓	SSIM ↑	FID ↓	mIOU ↑	FID ↓
T2I-Adapter	SD1.5	48.40	22.52	38.93	15.96	-	-
UniControlNet	SD1.5	40.65	22.27	41.55	17.14	-	-
UniControl	SD1.5	39.18	18.66	51.71	19.94	-	-
ControlNet	SD1.5	35.90	17.76	54.87	14.73	27.46	21.33
ControlNet++	SD1.5	<b>28.32</b>	16.66	57.06	18.23	<b>34.56</b>	19.29
EditAR (Ours)	LlamaGen	34.93	<b>15.97</b>	48.11	<b>13.91</b>	22.62	<b>16.13</b>

Table 2. Comparison of EditAR to various conditional image synthesis baselines. Our method yields the best FID scores, underscoring its strength in both sample quality and diversity. Note that T2I-Adaptor [39], ControlNet [74], and ControlNet++ [30] train separate models for different conditions. While UniControlNet [75] and UniControl [50] show promise as unified models capable of handling multiple tasks, they underperform EditAR. All the results are compared on 512 × 512 image resolution with Clean-FID implementation [46].

to-Prompt yields the strong editing clip scores, they fail to preserve the background (see also Figure 3). With further optimization, approaches like Null-text Inversion [38] and PnPInversion [28] show improved reconstruction fidelity. However, they remain limited by the absence of a unified, content-preserving model that performs consistently well across diverse tasks.

#### 4.3. Image Translation

We validate the effectiveness of the proposed method on three image translation tasks: segmentation-to-image, edge-to-image, and depth-to-image. As shown in Table 2 and Figure 4, EditAR outperforms other methods in FID across all tasks, demonstrating its ability to synthesize images that are both high-quality and diverse.

Compared with unified-model approaches (UniControl [50] and UniControlNet [75]), our method demonstrates superior scores. As evidenced by both approaches, unifying various conditional synthesis tasks alone is challenging. Specifically, UniControl employs a task-aware hyper-network to modulate the zero-convolution modules of Con-

trolNet. UniControlNet fine-tunes two additional adapters on frozen pre-trained text-to-image diffusion models, enabling various control inputs. Despite these advances, EditAR still delivers better overall performance.

Single-modality approaches take advantage of task-specific fine-tuning to achieve optimal performance on each domain. For example, in the segmentation-to-image task, single-model approaches often generate pre-defined classes, as they are trained solely to generate these specific categories. While EditAR consistently produces visually appealing results, the semantics may not always align perfectly with the ground-truth. Though learning a more challenging task, our model still synthesizes diverse images with good visual quality.

#### 4.4. Ablation

In this section, we study the effectiveness of the distillation loss, and show results distilled from different foundation models. Furthermore, we also explore the effects of different classifier-free-guidance, and show that it is crucial for high synthesis quality.



Figure 4. Visual comparisons to baseline methods on various image translation tasks. Our method, EditAR, produces photo-realistic results, preserves input details, and offers substantial sample diversity.

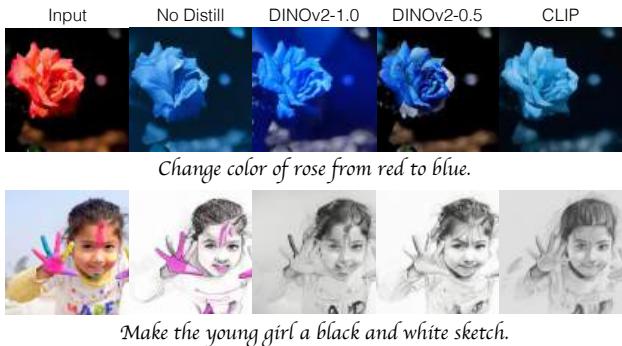


Figure 5. Studies on distillation loss. From left to right, we show the input image, results w/out distillation, and distillation results with DINOv2 and CLIP. The top example shows improved object localization. The bottom shows better text-to-image alignment.

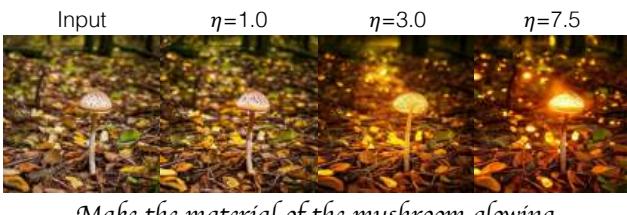


Figure 6. Ablation of classifier-free guidance values.

**Distillation.** We hypothesize that an autoregressive model trained solely to predict token indices is not guaranteed to align with semantic features. Therefore, we propose a distillation loss to encourage stronger feature space similarity between the autoregressive model and foundation models. As shown in Figure 5, experimental results show

that adding the distillation loss improves the overall text-to-image alignment, e.g., better localizing the target editing object. We try different foundation models and empirically find that DINOv2-0.5, with distillation coefficient  $\lambda_{distill} = 0.5$ , achieves the best overall performance.

**CFG Guidance.** We experiment with various guidance coefficients, as shown in Figure 6. Our results indicate that an optimal classifier-guidance strength is crucial for achieving high visual quality. With a lower classifier-free guidance value, the model shows weak responses to the text input. While a higher value improves text-to-image alignment, the reconstruction quality is reduced. In practice, we find that a value of  $\eta = 3$  yields the best trade-off between text-to-image alignment and reconstruction quality.

## 5. Conclusion

We introduce EditAR, a unified autoregressive framework designed for a wide range of conditional image generation tasks. EditAR adapts to various image inputs, switching modes solely through text prompts. We assess its effectiveness across diverse image editing and translation tasks, demonstrating its strong performance of reconstruction quality and edited visual quality. As text-to-image autoregressive models continue to advance, we hope this work paves the way for new possibilities in unified conditional generative modeling.

**Acknowledgements.** This project was supported, in part, by NSF CAREER Award IIS-2240014, NSF award IIS-2303153, and gifts from Qualcomm and Meta. We also acknowledge and thank the use of the Nautilus platform for the experiments discussed above.

## References

- [1] Anthropic. Claude. <https://www.anthropic.com/index/introducing-claude>, 2023. 3
- [2] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, 2(3):8, 2023. 2
- [3] Xia Bin, Wang Shiyin, Tao Yingfan, Wang Yitong, and Jia Jiaya. Llmga: Multimodal large language model based generation assistant. *ECCV*, 2024. 3
- [4] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, pages 18392–18402, 2023. 2, 5, 6, 7
- [5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3
- [6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020. 3
- [7] Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, pages 1209–1218, 2018. 5
- [8] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *ICCV*, pages 22560–22570, 2023. 2
- [9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9630–9640, 2021. 2, 5
- [10] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *CVPR*, pages 11305–11315, 2022. 2, 3
- [11] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 2
- [12] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pre-training from pixels. In *ICML*, pages 1691–1703. PMLR, 2020. 3
- [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. 3
- [14] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 34:8780–8794, 2021. 2
- [15] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pages 12873–12883, 2021. 2, 3
- [16] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yan-nik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. 2
- [17] Tsu-Jui Fu, Wenzhe Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding Instruction-based Image Editing via Multimodal Large Language Models. In *ICLR*, 2024. 3, 6, 7
- [18] 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, 2022. 2
- [19] Yuying Ge, Sijie Zhao, Chen Li, Yixiao Ge, and Ying Shan. Seed-data-edit technical report: A hybrid dataset for instructional image editing. *arXiv preprint arXiv:2405.04007*, 2024. 5
- [20] Google. Bard. <https://bard.google.com/>, 2023. 3
- [21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 3
- [22] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *ICLR*, 2023. 2, 3, 6
- [23] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2
- [24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 1
- [25] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022. 1, 2
- [26] Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *CVPR*, pages 8362–8371, 2024. 3
- [27] Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. Hq-edit: A high-quality dataset for instruction-based image editing. *arXiv preprint arXiv:2404.09990*, 2024. 5
- [28] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. *ICLR*, 2024. 2, 5, 6, 7
- [29] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023. 2

- [30] Ming Li, Taojiannan Yang, Huafeng Kuang, Jie Wu, Zhaoning Wang, Xuefeng Xiao, and Chen Chen. Controlnet++: Improving conditional controls with efficient consistency feedback. In *ECCV*, 2024. 2, 5, 7
- [31] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *CVPR*, pages 22511–22521, 2023. 3
- [32] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 3
- [33] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 3
- [34] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *NeurIPS*, 35:5775–5787, 2022. 2
- [35] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, pages 11461–11471, 2022. 2
- [36] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jianjun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2021. 2
- [37] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. *arXiv preprint arXiv:2305.16807*, 2023. 2
- [38] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, pages 6038–6047, 2023. 2, 7
- [39] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023. 3, 7
- [40] Jiteng Mu, Michaël Gharbi, Richard Zhang, Eli Shechtman, Nuno Vasconcelos, Xiaolong Wang, and Taesung Park. Editable image elements for controllable synthesis. In *ECCV*, pages 39–56, 2024. 2
- [41] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 1, 2
- [42] OpenAI. Introducing chatgpt. <https://openai.com/blog/chatgpt>, 2022. 3, 5
- [43] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 3
- [44] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 5
- [45] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *NeurIPS*, 35:27730–27744, 2022. 3
- [46] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *CVPR*, pages 11400–11410, 2022. 7
- [47] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH 2023, Los Angeles, CA, USA, August 6-10, 2023*, pages 11:1–11:11, 2023. 2
- [48] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023. 1, 2
- [49] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 2
- [50] Can Qin, Shu Zhang, Ning Yu, Yihao Feng, Xinyi Yang, Yingbo Zhou, Huan Wang, Juan Carlos Niebles, Caiming Xiong, Silvio Savarese, Stefano Ermon, Yun Fu, and Ran Xu. Unicontrol: A unified diffusion model for controllable visual generation in the wild. In *NeurIPS*, 2023. 3, 5, 7
- [51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3
- [52] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021. 1, 3
- [53] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 2
- [54] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *NeurIPS*, pages 14837–14847, 2019. 2, 3
- [55] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10674–10685, 2022. 1, 2
- [56] 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 *CVPR*, pages 22500–22510, 2023. 2
- [57] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghazemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep

- language understanding. *NeurIPS*, 35:36479–36494, 2022. [1](#) [2](#)
- [58] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. [1](#)
- [59] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 32, 2019.
- [60] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. [2](#)
- [61] 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. [2](#), [3](#), [4](#), [5](#)
- [62] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. [3](#)
- [63] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024. [2](#), [3](#)
- [64] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. [2](#), [3](#)
- [65] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikell, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korotnev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yunling Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. [2](#), [3](#)
- [66] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, pages 1921–1930, 2023. [5](#)
- [67] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, pages 6306–6315, 2017. [2](#), [3](#)
- [68] Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22532–22541, 2023. [2](#)
- [69] Navve Wasserman, Noam Rotstein, Roy Ganz, and Ron Kimmel. Paint by inpaint: Learning to add image objects by removing them first. *arXiv preprint arXiv:2404.18212*, 2024. [2](#), [5](#)
- [70] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *CVPR*, pages 18381–18391, 2023. [2](#)
- [71] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved VQGAN. In *ICLR*, 2022. [2](#)
- [72] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunnar Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. [2](#), [3](#)
- [73] Kai Zhang, Lingbo Mo, Wenhua Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *NeurIPS*, 36, 2024. [2](#)
- [74] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3836–3847, 2023. [2](#), [3](#), [7](#)
- [75] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K. Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. In *NeurIPS*, 2023. [3](#), [7](#)
- [76] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024. [3](#)

In this supplementary materials, we provide more details of the submission. We show additional editing results and more baselines in Section A complementing Section 4.2 in the paper; Furthermore, more image translation and comparisons are presented in Section B; More results of image translation with distillation are discussed in Section C. More implementation details and training recipes (paper Section 4.1) are discussed in Section D. *The models and codes will be released.*

## A. Additional Image Editing Comparison

In Table 1, Figure 3 and Section 4.2 in the main text, we have shown our editing results as well as comparisons to various baselines. Here we provide more details and show additional comparisons to more baselines, as shown in Table 3. More visual comparisons are presented in Figure 7, 8, 9, 10, 11. For all methods, we use their officially released model checkpoints to ensure quality. Note that in the paper where we referenced PnP Inversion, the method is also referred to as Direct Inversion. For consistency, we use PnP Inversion in the paper. We elaborate on the detailed implementations of each baseline below.

**InstructPix2Pix.** Instructional image editing poses greater challenges compared to text-to-image generation, as it usually requires the ability to process images following instructions while maintaining visual realism and reconstruction quality. InstructPix2Pix pioneered instruction-based image editing by creating a large-scale dataset comprising conditioning images, edited outputs, and corresponding editing instructions, then training a text-to-image diffusion model in a fully supervised manner. This method excels in tasks such as global texture transfer and object replacement. However, while InstructPix2Pix achieves a high edited CLIP score, it struggles to accurately reconstruct unedited regions, as observed by its lower whole CLIP score and background preservation scores. As visualized in Figure 7, 8, 9, 10, 11, results often reveal exaggerated edits and unrealistic modifications. To further evaluate if the differences are introduced by data alone, we fine-tuned InstructPix2Pix on the same datasets (SEED-DataEdit-Unsplash and PIPE Dataset) as our proposed method. Results show that the model only achieves an editing CLIP score of 19.97, with most examples failing to follow the given instructions. This shows the method is highly sensitive to input data, making it challenging to balance multiple datasets effectively.

**MagicBrush.** To enhance the editing quality of InstructPix2Pix, MagicBrush introduces a manually annotated dataset of 10K real image pairs (source image, instruction, target image) across diverse editing scenarios. As shown in Table 3, fine-tuning InstructPix2Pix on the MagicBrush dataset enhances reconstruction performance but leads to a significant decline in edited CLIP scores, high-

lighting the sensitivity of balancing data. Moreover, manual dataset annotation is time-consuming and challenging to scale efficiently.

**InstructDiffusion.** InstructDiffusion aims to develop a unified model capable of addressing a wide range of vision tasks without requiring task-specific modifications. To enable this, it extends InstructPix2Pix by training on diverse tasks, including understanding tasks (e.g., segmentation and keypoint detection) and generative tasks (e.g., editing and enhancement). The approach consists of two stages: training a unified model across various tasks, followed by fine-tuning for specific tasks, similar to InstructPix2Pix. Specifically, for image editing, the paper introduces a new dataset, Image Editing in the Wild (IEIW), created by combining multiple existing datasets. However, as shown in Figure 7, 8, 9, 10, 11, it often produces exaggerated editing results.

**MGIE.** MLLM-Guided Image Editing (MGIE) highlights that the reliance on CLIP text encoders in Stable Diffusion limits the ability to follow precise instructions for achieving specific editing goals. To address this, MGIE replaces the CLIP text encoder with outputs from multimodal large language models (MLLMs), enabling the understanding of more expressive and detailed instructions. In contrast to MGIE, our method does not rely on a diffusion model, resulting in significantly lower complexity and avoiding the challenges of jointly optimizing models. As shown in Table 3, despite its simplicity, our approach delivers significantly better reconstruction and editing quality.

**SEED-X-Edit.** SEED-X is a unified and versatile model that can handle both comprehension and generation tasks, showcasing strong performance in real-world applications across various domains, e.g., instructed image editing. SEED-X-Edit refers to the model derived by fine-tuning SEED-X specifically for image editing on the SEED-DataEdit dataset, a new dataset containing both manual annotated data and automatically generated image pairs. As shown in Table 1 and Figures 7, 8, 9, 10, 11, the method struggles with reconstructing unedited regions and often produces unrealistic images with high contrast. In comparison, our proposed method achieves better overall performance, despite being trained without manual-annotated data and with a simpler design.

**Inversion-based Methods.** Unlike feed-forward methods, inversion-based approaches first invert an image into a latent space, typically using latent noise or embeddings, before performing content-preserving sampling to generate the edited target. These methods require not only an editing instruction but also a source prompt, which is crucial for the inversion process. For example, Prompt-to-Prompt uses DDIM inversion and manipulates attention maps to preserve content across various edits. Similarly, Pix2Pix-Zero retains the cross-attention maps of the input image throughout the diffusion process for improved reconstruc-

Method	T2I Model	Structure Distance ↓	Background Preservation				CLIP Similarity	
			PSNR ↑	LPIPS ↓	MSE ↓	SSIM ↑	Whole ↑	Edited ↑
Prompt-to-Prompt	SD1.4	69.43	17.87	208.80	219.88	71.14	25.01	22.44
Null-text Inversion	SD1.4	13.44	27.03	60.67	35.86	84.11	24.75	21.86
PnP Inversion	SD1.4	11.65	27.22	54.55	32.86	84.76	25.02	22.10
Pix2Pix-Zero	SD1.4	61.68	20.44	172.22	144.12	74.67	22.80	20.54
MasaCtrl	SD1.4	28.38	22.17	106.62	86.97	79.67	23.96	21.16
InstructPix2Pix	SD1.5	107.43	16.69	271.33	392.22	68.39	23.49	22.20
MagicBrush	SD1.5	26.81	26.85	66.67	171.11	83.37	23.89	20.84
InstructDiffusion	SD1.5	74.21	20.88	142.35	353.45	76.70	24.06	21.57
MGIE	SD1.5	67.41	21.20	142.25	295.11	77.52	24.28	21.79
SEED-X-Edit	SD-XL	61.69	18.80	173.63	209.05	74.93	25.51	22.20
EditAR (Ours)	LlamaGen	39.43	21.32	117.15	130.27	75.13	24.87	21.87

Table 3. Comparisons complementing Table 1. Comparison of EditAR to various feed-forward methods (bottom) and inversion-based approaches (top) on the PIE-Bench dataset. Our results attain superior results in preserving the details of the input as well as following the new edits, narrowing the gap with advanced inversion-based methods. The feed-forward baseline results show various types of failures, such as decline in image quality, unfaithful background preservation, and not following the editing instructions. While InstructPix2Pix achieves a high edited CLIP score, it struggles to reconstruct unedited regions accurately, as indicated by its lower whole CLIP score and background scores. MagicBrush shows improved background preservation but at the expense of editing quality. InstructDiffusion and MGIE shows improved reconstruction and editing quality, yet our method demonstrates stronger overall performance. Seed-X-Edit struggles with reconstructing unedited regions and produces images with unrealistic contrast.

tion. However, compared to our approach, these methods struggle to preserve the background. To enhance non-rigid editing, MasaCtrl modifies self-attention in diffusion models into mutual self-attention, enabling effective blending of local content and textures from input images during generation. While specialized for non-rigid edits, it falls short when applied to a variety of editing tasks. As indicated by the CLIP similarity, our method achieves better overall responsiveness to edits.

For improved reconstruction quality, optimization-based inversion methods like Null-text Inversion and PnP Inversion are proposed to invert the conditioning image into a latent embedding, achieving near-perfect reconstruction. Note the numbers in Table 3 for both methods are produced with Prompt-to-Prompt. Although these methods generate higher-quality visuals, they require additional computation and time to optimize the latent embeddings. As shown in Figures 7, 8, 9, 10, 11, these methods are still limited by the lack of a unified, content-preserving model that consistently performs well across diverse tasks, thus limiting their scope.

## B. Additional Image Translation Comparison

In Table 2, Figure 4 and Section 4.3 in the main text, we have shown our image translation results as well as comparisons to various baselines. Here we provide more details show more visual comparisons: depth-to-image in Figure 12, edge-to-image in Figure 13, segmentation-to-image in Figure 14. ControlNet results are produced with Con-

trolNet v1.1. For UniControlNet, Unicontrol and ControlNet++, we use their official released checkpoints. For each method, we produce 5,000 examples of resolution  $512 \times 512$  to measure the corresponding metrics. Results show that, though learning a more challenging task, our model still synthesizes diverse images with good visual quality.

## C. Distillation

In Section 4.4 of the main text, we qualitatively show that adding the distillation loss improves the overall text-to-image alignment, e.g., better localizing the target editing object, on the task of image editing. For image translation, our results show that the FID scores are improved from 16.35 to 15.97 for depth-to-image, 14.43 to 13.91 for edge-to-image, 16.52 to 16.13 for segmentation-to-image. These results further emphasize the importance of enforcing a stronger feature space similarity between the autoregressive model and foundation models, leading to models with stronger performance across tasks.

## D. Implementation Details

**Evaluation and Metrics.** For image editing, the PIE-Bench dataset is used for evaluation. Specifically, PIE-Bench contains 700 images featuring ten distinct editing types: (0) random editing, (1) change object, (2) add object, (3) delete object, (4) change object content, (5) change object pose, (6) change object color, (7) change object material, (8) change background, and (9) change image style. Within

each scene, images are evenly distributed among four categories: animal, human, indoor environment, and outdoor environment. Our method as well as all other feed-forward methods uses the source image and editing instructions to predict the target edit. The inversion-based approaches use the source image, the source prompt, and the target image prompt. Structure Distance ( $\times 10^3$ ) leverages self-similarity of DINO-ViT features and computes cosine similarity between image features as structure distance. PSNR, LPIPS ( $\times 10^3$ ), MSE ( $\times 10^4$ ), and SSIM ( $\times 10^2$ ) are reported to compare the background preservation using the manual-annotated masks. The CLIP score ( $\times 10^2$ ) evaluates text-image similarity between the edited images and corresponding target editing text prompts. Both the whole image and regions in the editing mask (black out everything outside the mask) are calculated, and referred to as Whole Image Clip and Edit Region Clip, respectively. All metrics are computed at the resolution of  $512 \times 512$ .

For the evaluation of image translation, we follow ControlNet++ and use the corresponding validation splits for COCOSTuff and MultiGen-20M, which contain 5,000 examples per task. Regarding metrics, we follow the common practice in the field: mIOU ( $\times 10^2$ ) is used for semantic segmentation conditions, RMSE for depth map conditions, and SSIM ( $\times 10^2$ ) for canny edge conditions. FID scores are computed with 5,000 images at the resolution of  $512 \times 512$ .

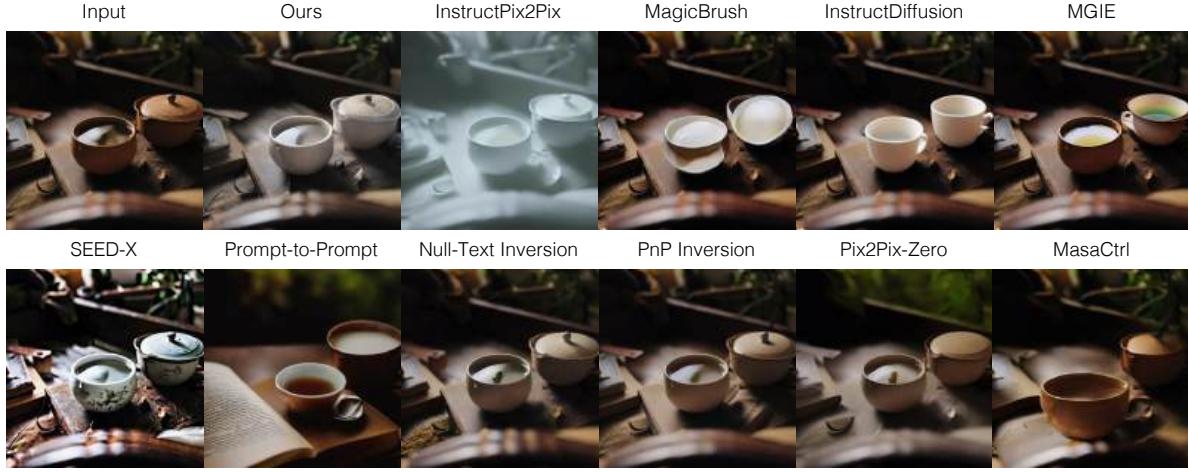
**Training and Inference.** To overcome varying imbalances between tasks, datasets must be mixed thoughtfully. We mix datasets by sampling 15% for each image translation task, 25% for PIPE dataset, and 30% for SEED-Data-Edit-Unsplash. The training hyperparameters mostly follow LlamaGen. All images are resized to a resolution of  $512 \times 512$  for both training and inference. The VQ-Autoencoder has a downsampling ratio of 16, so that each image is represented by 1024 tokens. Its dictionary size is 16384 and embedding dimensionality is 8. The text encoder utilizes Flan-T5-XL, producing a sequence of 120 embeddings. We use the pre-trained text-to-image autoregressive model LlamaGen GPT-XL, which has 36 layers and an embedding dimension of 1280. The model is optimized using AdamW with a constant learning rate of  $10^{-4}$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.95$ , and weight decay of 0.05. The model is trained with a batch size of 64 for 40,000 iterations on 8 A100 GPUs. We use  $\lambda_{distill} = 0.5$  and  $\eta = 3.0$  for inference.

## E. Discussion

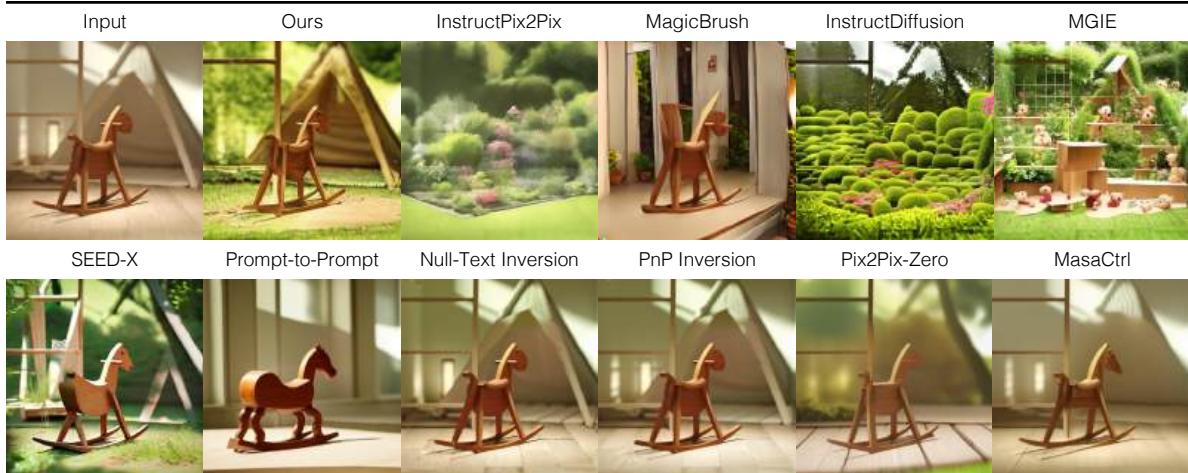
EditAR is a versatile autoregressive model that unifies multiple conditional image generation tasks within a single framework. Using only text prompts, the model seamlessly adapts to various image inputs and tasks. Our comprehensive evaluation demonstrates EditAR’s exceptional performance in both image editing and diverse image translation tasks. This work represents a significant milestone as the

first demonstration that a single autoregressive model using next-token prediction can effectively handle various conditional generation tasks on large-scale benchmarks. By successfully tackling multiple conditional image generation challenges, EditAR opens new possibilities for unified conditional generation approaches, building upon recent advances in text-to-image autoregressive modeling.

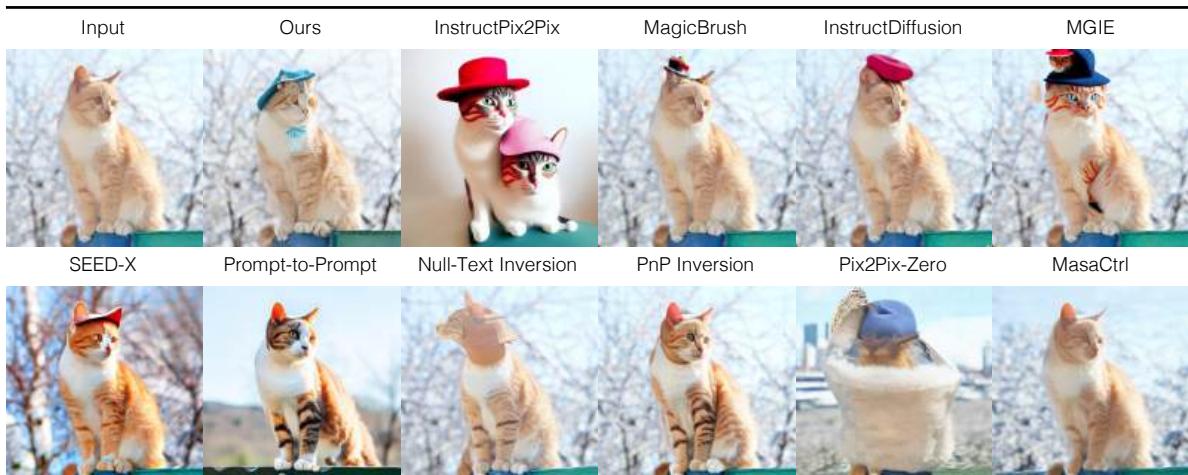
**Limitations.** EditAR builds upon autoregressive text-to-image models, allowing it to naturally benefit from advances in base model quality. Besides, the current implementation is restricted to single-image conditional inputs, though the framework could theoretically handle multiple conditions. Additionally, the model struggles with non-rigid or 3D editing tasks due to the insufficient training data. Addressing these challenges through expanded datasets and architectural enhancements represents an important direction for future research.



*Change the color of the tea cup to white.*

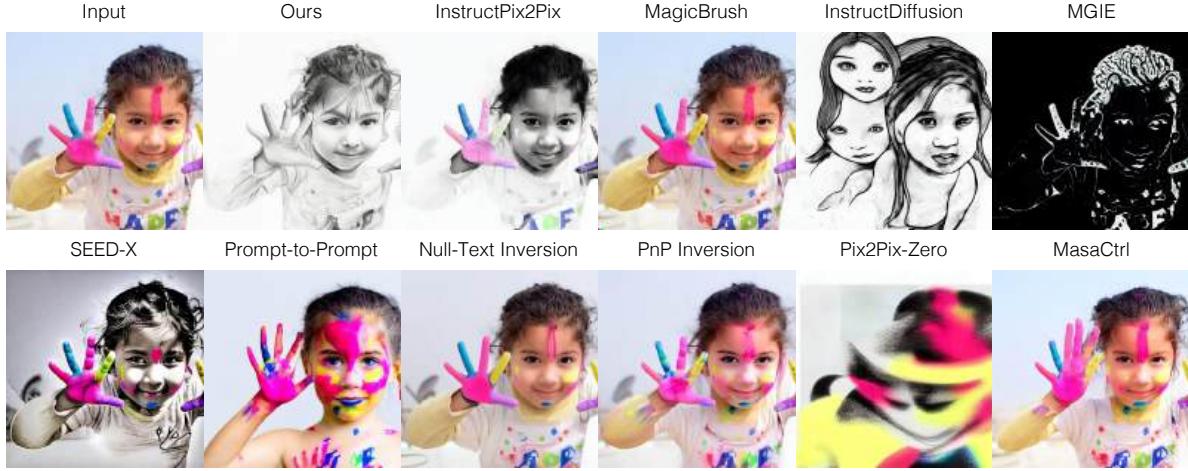


*Change the room to a garden.*



*Make the cat wear a hat.*

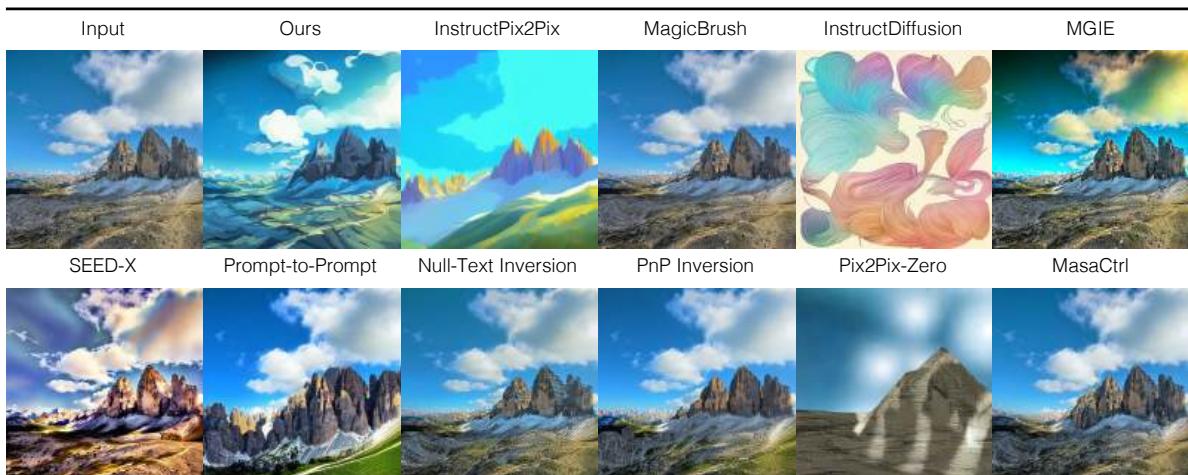
Figure 7. Comparisons complementing Figure 3. Comparison of EditAR to various feed-forward methods and inversion-based approaches on the PIE-Bench dataset. Our results attain superior results in preserving the details of the input as well as following the given edits.



*Make the young girl a black and white sketch.*



*Make the camera a wooden toy.*

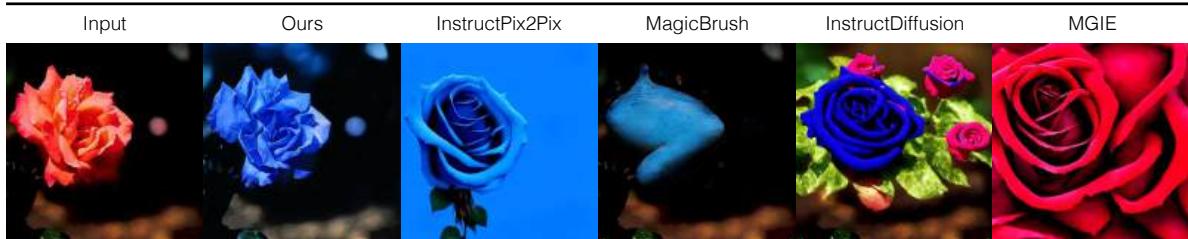


*Create digital art style.*

Figure 8. Comparisons complementing Figure 3. Comparison of EditAR to various feed-forward methods and inversion-based approaches on the PIE-Bench dataset. Our results attain superior results in preserving the details of the input as well as following the given edits.



*Remove the husky dog.*



*Change the color of the rose from red to blue.*

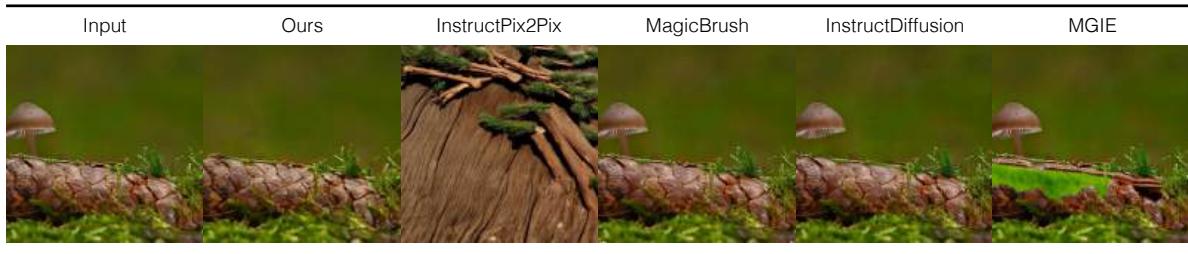


*Change the kitten's color to yellow.*

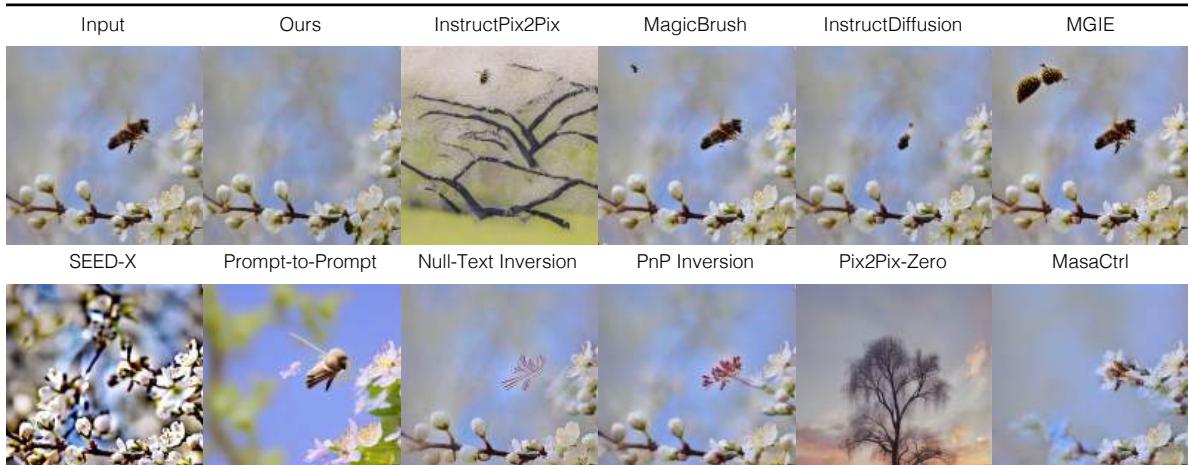
Figure 9. Comparisons complementing Figure 3. Comparison of EditAR to various feed-forward methods and inversion-based approaches on the PIE-Bench dataset. Our results attain superior results in preserving the details of the input as well as following the given edits.



*Change the season from autumn to spring.*

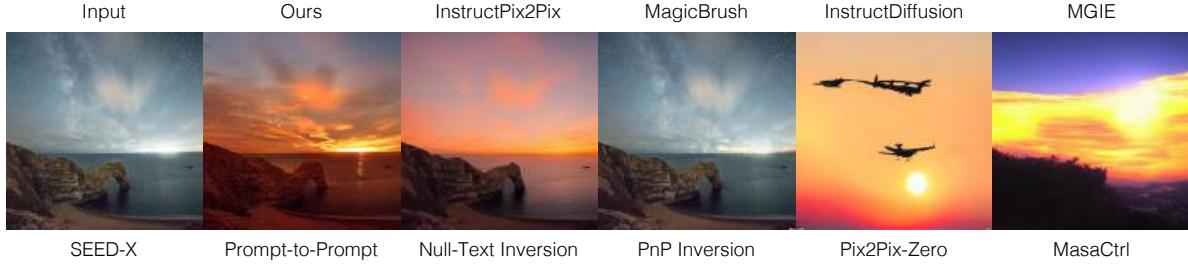


*Remove the small mushroom from the pine branch.*



*Remove the bee flying over the flowering tree branch.*

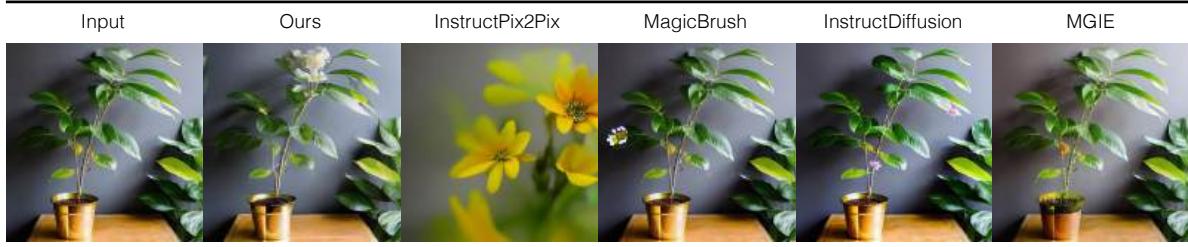
Figure 10. Comparisons complementing Figure 3. Comparison of EditAR to various feed-forward methods and inversion-based approaches on the PIE-Bench dataset. Our results attain superior results in preserving the details of the input as well as following the given edits.



*Change the state of the sky to sunset.*



*Change the environment from a field to a forest.*



*Change the plant to a flower.*

Figure 11. Comparisons complementing Figure 3. Comparison of EditAR to various feed-forward methods and inversion-based approaches on the PIE-Bench dataset. Our results attain superior results in preserving the details of the input as well as following the given edits.

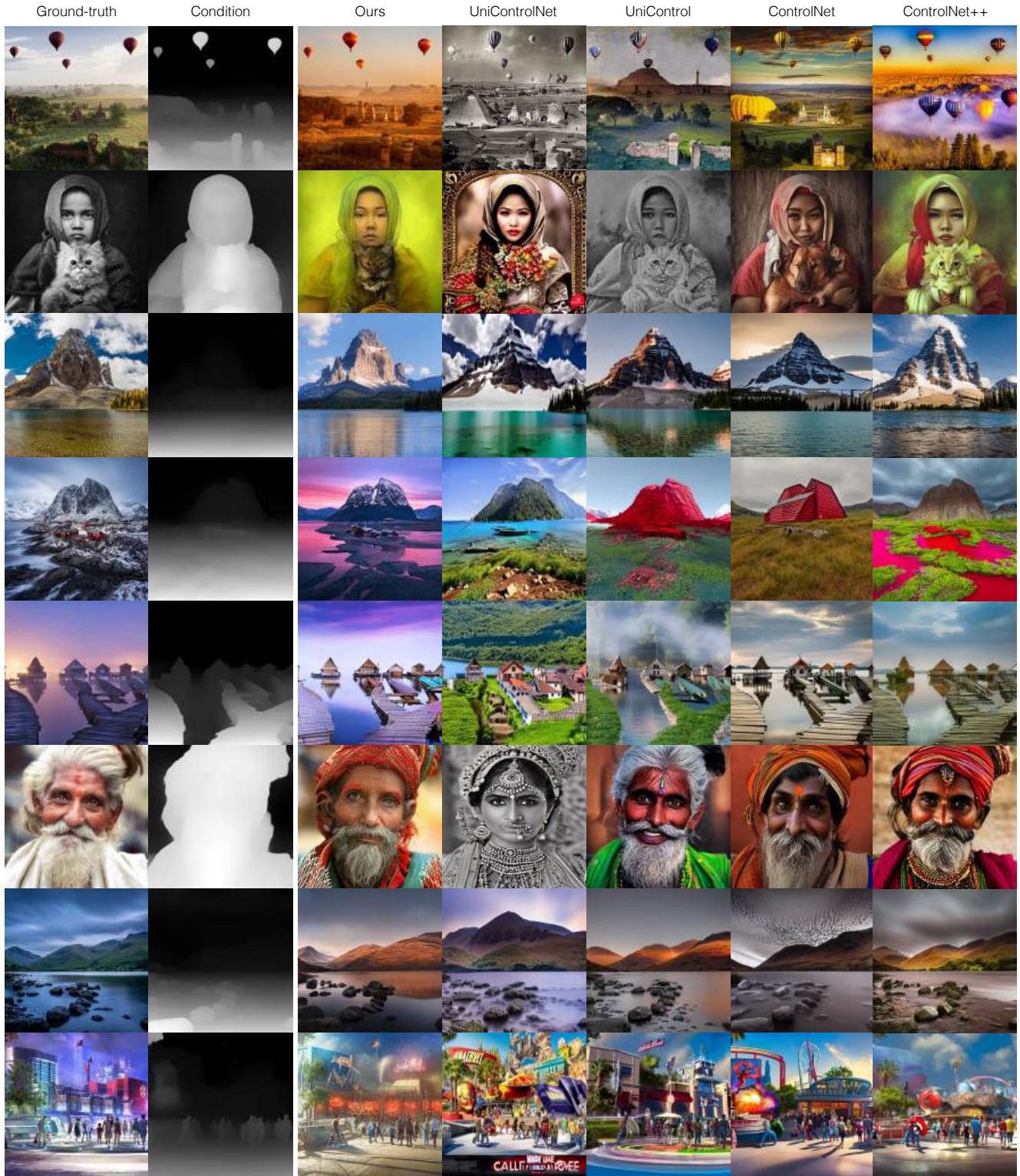


Figure 12. Comparisons complementing Figure 4. Visual comparisons to baseline methods on various depth-to-image translation. Our method, EditAR, produces photo-realistic results, preserves input details, and offers substantial sample diversity.

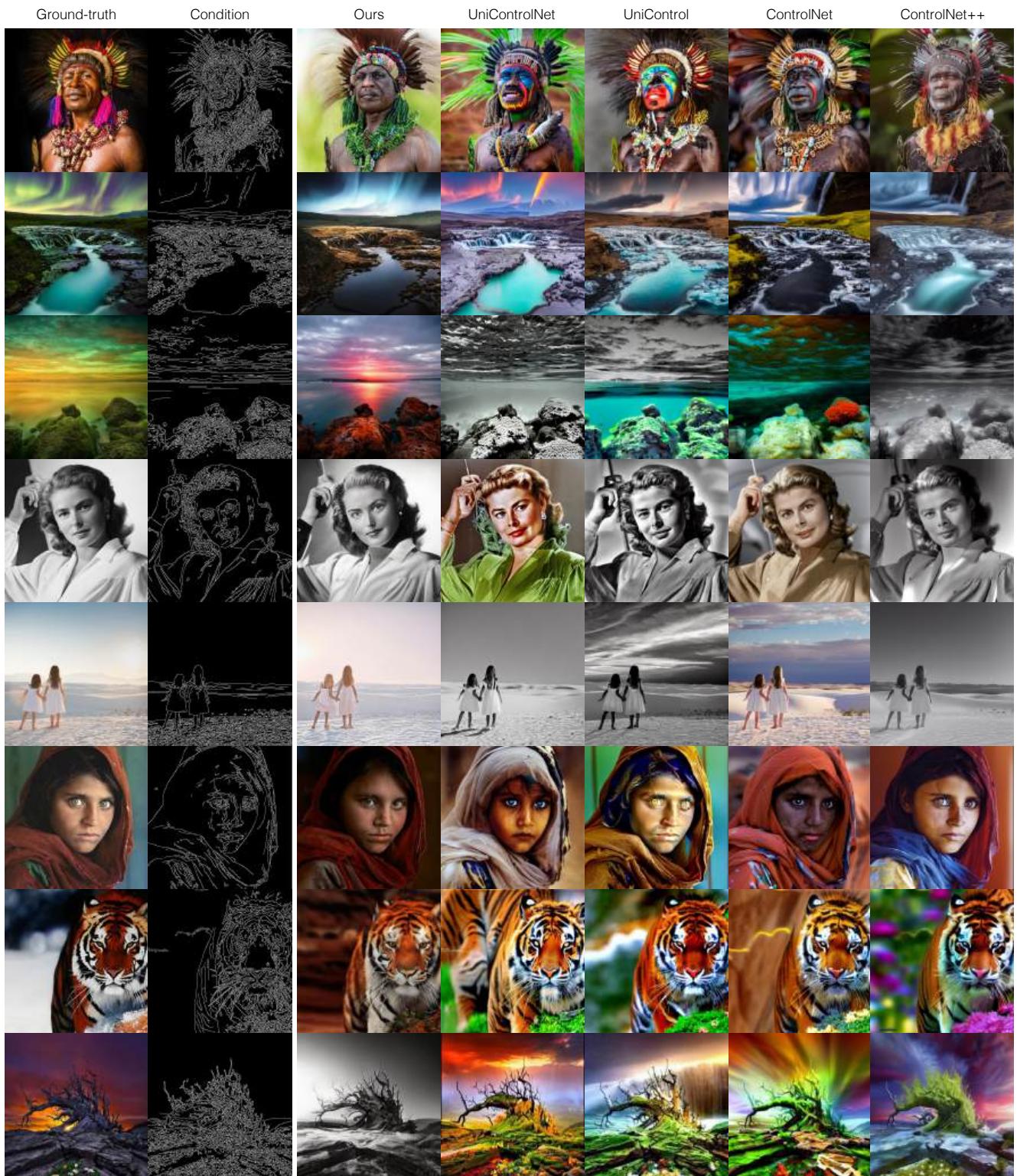


Figure 13. Comparisons complementing Figure 4. Visual comparisons to baseline methods on various edge-to-image translation. Our method, EditAR, produces photo-realistic results, preserves input details, and offers substantial sample diversity.



Figure 14. Comparisons complementing Figure 4. Visual comparisons to baseline methods on various segmentation-to-image translation. Our method, EditAR, produces photo-realistic results, preserves input details, and offers substantial sample diversity.