EditAR: Unified Conditional Generation with Autoregressive Models Supplementary Materials

Jiteng Mu[1], Nuno Vasconcelos[1], Xiaolong Wang[1,2]
[1]UC San Diego, [2]NVIDIA

In this supplementary materials, we provide more details of the submission. We show additional editing results and more baselines in Section 1 complementing Section 4.2 in the paper; Furthermore, more image translation and comparisons are presented in Section 2; More results of image translation with distillation are discussed in Section 3. More implementation details and training recipes (paper Section 4.1) are discussed in Section 4.

1. 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 1. More visual comparisons are presented in Figure 1, 2, 3, 4, 5. 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 1, 2, 3, 4, 5, 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-Data-Edit-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 1, fine-tuning InstructPix2Pix on the MagicBrush dataset enhances reconstruction performance but leads to a significant decline in edited CLIP scores, highlighting 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 1, 2, 3, 4, 5, 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 1, 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,

Method	T2I Model	Structure Distance ↓	Background Preservation PSNR ↑	LPIPS ↓	MSE ↓	SSIM ↑	CLIP Similarity 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 1. 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.

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-Data-Edit dataset, a new dataset containing both manual annotated data and automatically generated image pairs. As shown in Table 1 and Figures 1, 2, 3, 4, 5, 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 reconstruction. 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 1 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 1, 2, 3, 4, 5, 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.

2. 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 6, edge-to-image in Figure 7, segmentation-to-image in Figure 8. ControlNet results are produced with ControlNet v1.1. For UniControlNet, Unicontrol and ControlNet++, we use their official released checkpoints. For each method, we produce 5,000 examples of resolution 512×512 to measure the corresponding metrics. Results show that, though learning a more challenging task, our model still synthesizes diverse images with good visual quality.

3. 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.

4. 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 512×512.

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 512×512.

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 512×512 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.

5. 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.

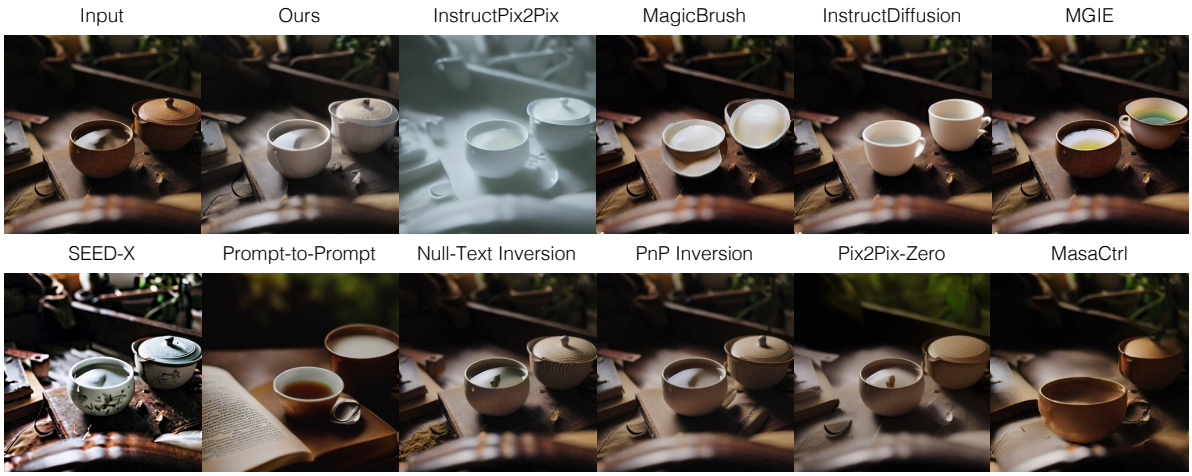

Figure 1. Comparisons complementing Figure 3. Comparison of EditAR to various feed-forward methods and inversion-based approaches on the PIE-Bench dataset. Our results attain superior results in preserving the details of the input as well as following the given edits.

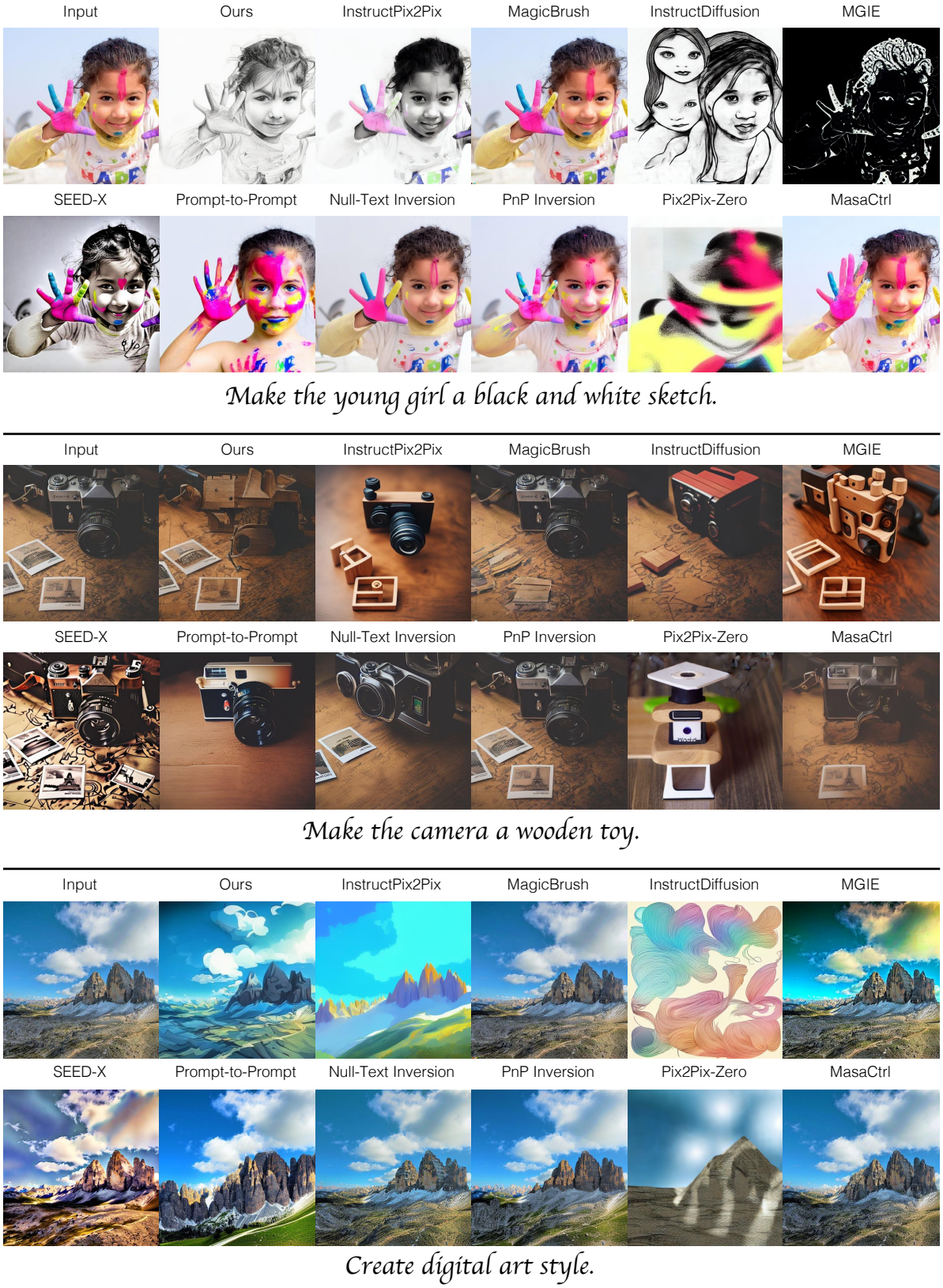

Figure 2. Comparisons complementing Figure 3. Comparison of EditAR to various feed-forward methods and inversion-based approaches on the PIE-Bench dataset. Our results attain superior results in preserving the details of the input as well as following the given edits.

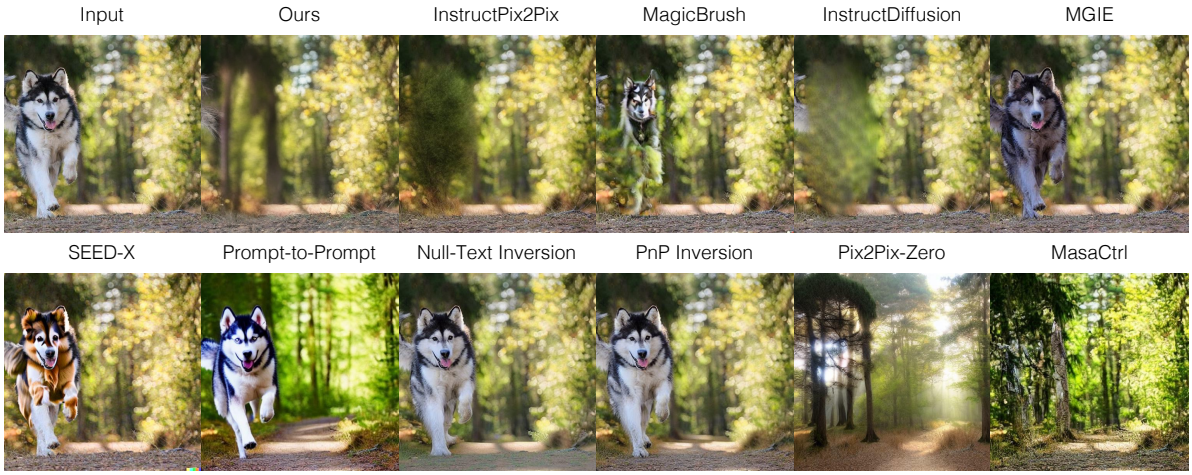

Figure 3. Comparisons complementing Figure 3. Comparison of EditAR to various feed-forward methods and inversion-based approaches on the PIE-Bench dataset. Our results attain superior results in preserving the details of the input as well as following the given edits.

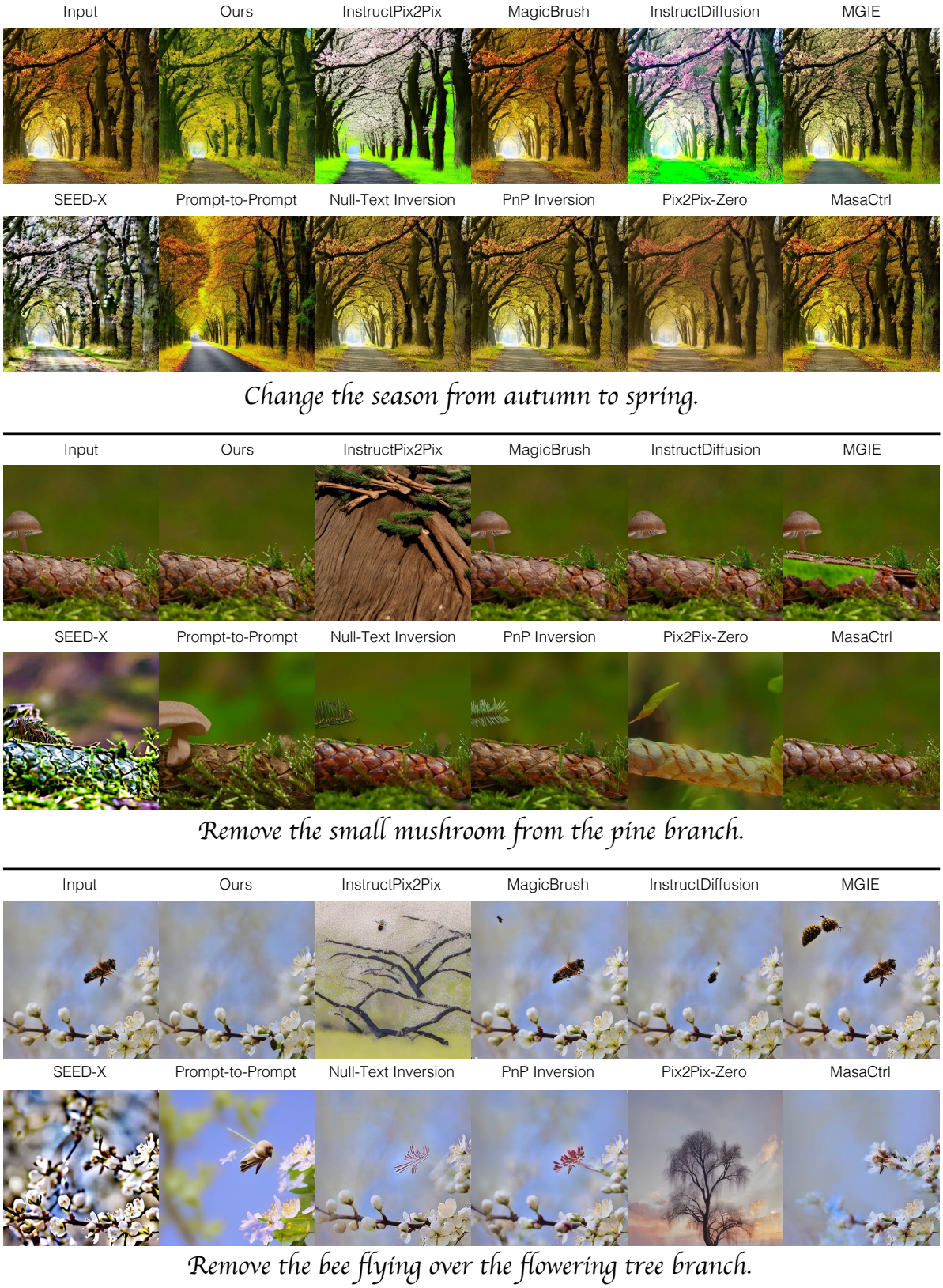

Figure 4. Comparisons complementing Figure 3. Comparison of EditAR to various feed-forward methods and inversion-based approaches on the PIE-Bench dataset. Our results attain superior results in preserving the details of the input as well as following the given edits.

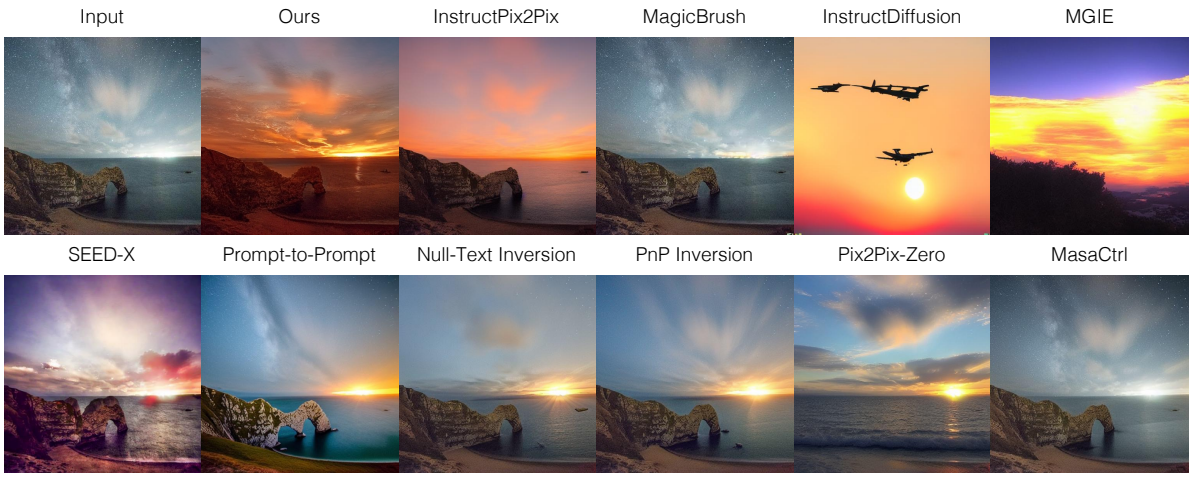

Figure 5. Comparisons complementing Figure 3. Comparison of EditAR to various feed-forward methods and inversion-based approaches on the PIE-Bench dataset. Our results attain superior results in preserving the details of the input as well as following the given edits.

Figure 6. 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 7. 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 8. 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.