X-Prompt: Generalizable Auto-Regressive Visual Learning with In-Context Prompting

Supplementary Material

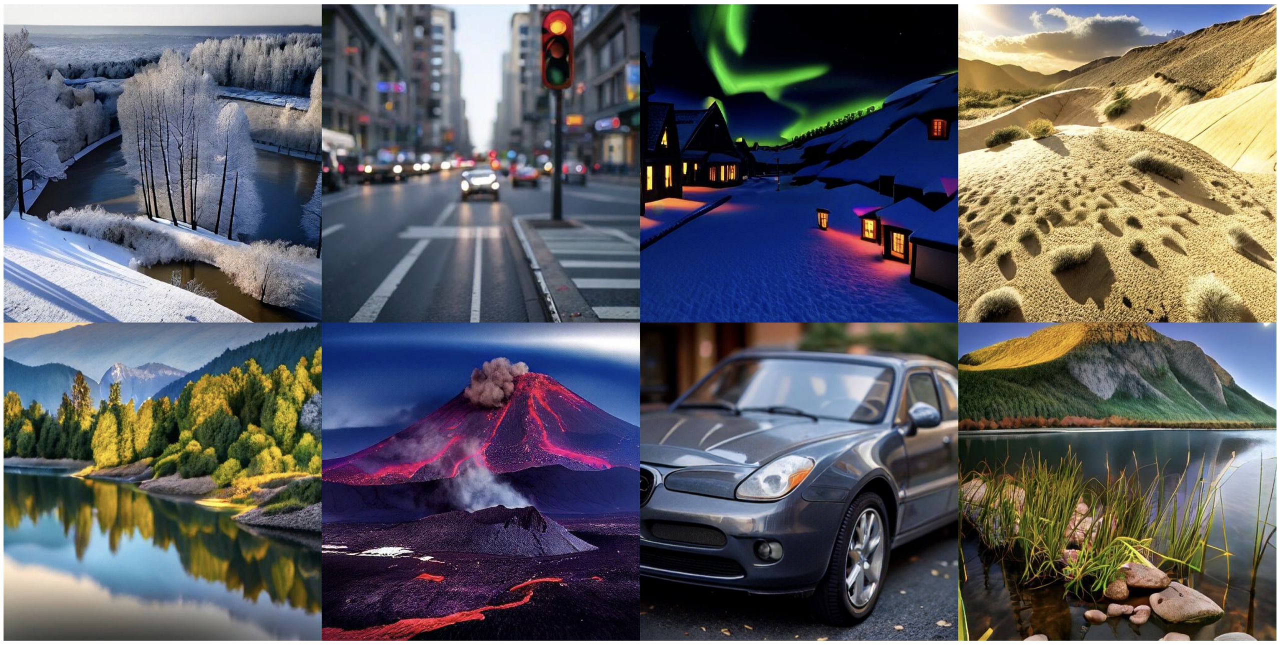


Figure 7. **Qualitative results of text-to-image generation.** High-quality text-to-image generation cases with high aesthetic qualities after training on Laion-Aesthetics [49].

A. Qualitative Results of Text-to-Image Generation.

We visualize some text-to-image generation results of our model in Fig. 7 and comparison with other models in Fig. 11. Fig. 7 demonstrates that our model can generate images with high aesthetic qualities after training on filtered high qaulity data from Laion-Aesthetics [49]. Figure 11 clearly demonstrates that the incorporation image dense description task has significantly bolstered the model's proficiency in accurately following text prompt when compared to other models such as Emu3 [69] and Janus [71].

B. Details of model architecture.

As illustrated in Fig. 8, the forward process of X-Prompt can be decoupled into two distinct stages during both training and inference.

During training, the attention mechanism is split into two parts, effectively doubling the context length. This extension is crucial, as Chameleon is pre-trained with a 4K context length, which is insufficient for in-context learning. The split attention mechanism preserves two complete staircase-shaped attention patterns, enabling full utilization of efficient training acceleration libraries such as FlashAttention [11]. The X-Prompt tokens, which are task-agnostic and learnable during training, remain fixed during inference.

At inference time, Stage 1 compresses the In-context Example (IE) tokens into X-Prompt (XP) tokens. In Stage 2, the IE tokens are removed from the key-value (KV) cache, allowing the model to perform X-Prompt-guided generation efficiently. This KV cache management strategy aligns well with the requirements of high-speed inference, ensuring computational efficiency and reduced memory overhead.

C. Model efficiency analysis.

As shown in Fig. 8. At inference time. In stage1, In-context Example (IE) tokens are compressed into XP tokens. In stage2, IE tokens are removed from KV cache. X-Prompt tokens and its hidden states stay in KV cache for downstream tasks. At inference time. In stage1, In-context Example (IE) tokens are compressed into X-Prompt tokens. This process reduces 56.1% FLOPs when conducting transformation on a single image from an In-context example

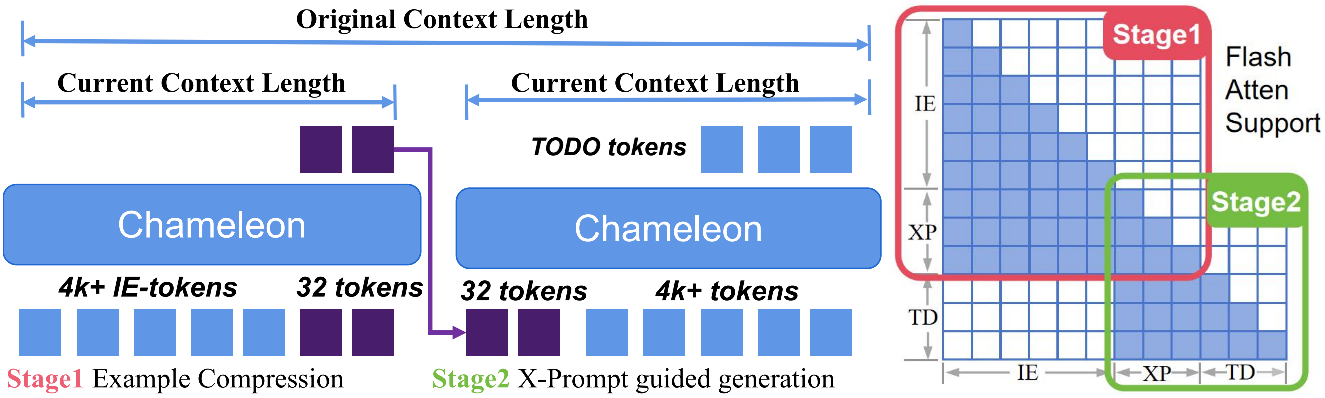


Figure 8. **Two stages of X-Prompt in both training and inference.** In stage 1, in-context examples are compressed into X-Prompt tokens. In stage 2. these tokens (with their hidden states) serves as task guidance for the specified tasks.

and even more when transforming multiple image from the same in-context examples. In addition to efficiency, as reported in Tab. 4 and Tab. 5 in the main paper, the X-Prompt process can also improve the performance of in-context learning by introducing feature compression strategy.

D. Ablation on number of X-Prompt tokens.

Reported in Tab. 6, we set the number of X-Prompt tokens to 32, as smaller number leads to significant performance degradation while increasing the number beyond this does not provide additional gains.

E. Higher Resolution Reconstruction

F. User Preference Study.

Table 7 reports the user study of the X-Prompt's winning rate with 41 volunteers of college students and ordinary people on image quality and text alignment over 400 generated images with prompt from GenEval [25] and 300 edit images from MagicBrush [81] testset. X-Prompt can achieve text-to-image generation quality on par with SDXL [42] with image editing quality surpass [81].

G. Details of training data

Full training data statistics are reported in Tab. 10 and Tab. 11. For each of the task in Tab. 10, we use QWenVL-2 [66] to describe the transformation between the input and output images and augments with reversed task.

H. Details of Prompt template

H.1. Difference description task.

For each image editing pair in Ultra-Edit [85] and MagicBrush [81], we leverage QWenVL2 [66] to describe the difference between images using the following prompt: "Describe the differeces between two images. Use 'input image' describe the first image and 'output image' to describe the second image, describe what subtask it belongs to, choosing from [Style Transfer, Object Removal, Object Replacement, Object Addition, Object Modification, Season/ Time Change, OTHER_SUBTASK]". We also ask QWenVL2 to label a reverse editing prompt for data augmentation.

H.2. filtering data generated by RB-Modulation.

To generate and filter high-quality data, we first use FLUX to generate high-quality and stylized images based on the prompt templates in Tab. 12. However, RB-Modulation [48] occasionally performs correct style transformations but sometimes fails. To ensure quality, we further use QWen-VL2 [66] for data filtering. Due to QWen-VL2's current limitations in analyzing relationships among three images, we conduct quality filtering in two stages. First, we ask QWenVL-2 to verify the consistency of the main object and semantic with the base image using the question: "Do you think the two image shares the same semantics and basic layout [Yes/No]? Provide your reasoning.". Next, we check the success of style transfer from exemplar image by asking "Do you think the two image shares the same style [Yes/No]? Provide your reasoning." Through this process, we filter 10K high quality image based style-personalization image pairs to incorporate into the training of X-Prompt.

H.3. filtering data generated by IP-Adapter.

IP-Adapter [79] can perform layout and semantic combination on two provided images, However, the final output image can maintain different attributes (layout, semantics, texture, details) from different images in a unified but not entirely deterministic format. Given the complex attributes relationship between the input images and the output images, we employ GPT-4o to analyze and annotate these relationships. As shown in Fig. 10, GPT-4o provides high-

Number of XP tokens	2	8	32	64
Object Add CLIP$_{\text{dir}}$	-0.012	0.053	0.092	0.089
Image Enhancement PSNR	8.04	13.50	17.22	17.25

Table 6. Results of models trained with different #XP-tokens

Domain	Model	Alignment	Quality
Text2Image	SD-1.5 [51]	69.33%	55.40%
	SDXL [46]	58.60%	49.31%
	Chameleon [62]	70.42%	80.19%
	Emu3Gen [69]	57.13%	55.25%
Image Editing	MagicBrush [81]	56.74%	51.37%

Table 7. User study of X-Prompt winning rate with 41 volunteers of college students and ordinary people on image quality and text alignment over 400 generated images with prompt from GenEval [25] and 300 edit images from MagicBrush [81] testset.

Model	Resolution	PSNR	SSIM
SDXL-VAE (16x)	512	27.51	0.810
	1024	32.13	0.922
Chemeleon-VQVAE (16x)	512	26.34	0.805
	1024	29.77	0.906
Emu3-VQVAE (8x)	512	27.78	0.833

Table 8. **Reconstruction quality tested on Rain-100L**. Increasing resolution can greatly enhance reconstruction quality.

quality, detailed descriptions of the relationships between different images. For this purpose, we annotate a dataset of 50K image pairs.

H.4. Caption Rewriting on Laion-aesthetic.

We filter high-quality data from Laion-Aesthetic [49], selecting images with an aesthetic score greater than 6. For dense caption rewriting, we use QWen-VL2 [66], focusing on the relative positions, colors, and numbers of objects. To preserve caption diversity, we retain 10% of the original captions during training.

I. Retrieval-Augmented Image Editing

Clustering similar editing pairs during training is critical to the success of Retrieval-Augmented Image Editing (RAIE) as a form of in-context learning. Fortunately, we observe that many editing instructions in MagicBrush [81] and UltraEdit [85] are highly similar to each other. As shown in Fig. 12, by pairing each editing pair with its nearest neighbor based on CLIP [44] text feature similarity, we find that many instructions are either similar or identical. This similarity is a key factor contributing to the effectiveness of RAIE.

J. More Qualitative Results Visualization.

We provide more visualization on vision tasks in Fig. 9.

K. Upper Bound Analysis on Low Level Vision Tasks.

As reported in Tab. 9 with “model prediction (upper bound)” format. “Upper bound” is tested with ground truth image directly go through VAE or VQ-VAE for direct reconstruction while “model prediction” is tested on model predicted output. As the performance of X-Prompt is constrained by VQ-VAE of Chameleon [59] model. We believe orthogonal researches on improving VQ tokenizer will significantly improve the performance in the near future.

We use the Rain-100L derained test set to evaluate the reconstruction abilities of different models. As shown in Tab. 8, increasing the input resolution significantly enhances reconstruction quality for both VQ-VAE and VAE models. This improvement arises from the fact that image compression inherently leads to a loss of detail, and providing higher-resolution input allows the model to recover previously lost details, resulting in better outputs. However, we are unable to implement X-Prompt with a 1024 resolution as Chameleon [59] is pretrained exclusively on a 512 resolution. We anticipate significant improvements across all tasks with the availability of higher-resolution early-fusion multi-modal foundation models in the future.

Image Representation	ADE-20K mcIoU	GoPro PSNR	SSID PSNR	LOL PSNR	Rain100H PSNR
InstructDiffusion [21] (CVPR-2024) SD-VAE	33.62 (50.05)	23.58 (29.54)	34.26 (36.56)	-	-
Chemelon [59] VQVAE	36.71 (43.44)	21.04 (28.58)	31.91 (33.35)	19.71 (22.27)	24.77 (27.12)

Table 9. **Upper bound analysis** reported in “model prediction (upper bound)” format. “Upper bound” is tested with ground truth image directly go through VAE or VQ-VAE for direct reconstruction while “model prediction” is tested on model predicted output. As the performance of X-Prompt is constrained by VQ-VAE of Chameleon [59] model. We believe orthogonal researches on improving VQ tokenizer will significantly improve the performance in the near future.

	DFWB	GoPro	Rain13k	mit5k	LoL	Laion_Aesthetic	Ultra-Edit	MagicBrush	NYU-v2-depth	ADE20K	ScannNet-Norm	dep/seg/norm/hed/mlsd2img
Ori_data	72K	17K	13K	5K	6K	500K	500K	1.7K	48K	20K	260K	100K × 5
Augmentation	288K	68K	52K	20K	24K	1000K	2000K	6.8K	192K	80K	1040K	100K × 20

Table 10. **Detailed statistics of training data with augmentation.** For each pair, we use reverse task and difference description task to augment the data.

	RB-Modulation	IP-Adapter	Viton-Try-On	Pose&Action	MimicBrush
Ori_data	10K	50K	120K	10K	50K

Table 11. **Detailed statistics of training data without augmentation.**

Rendering	Sketch	Poster
Mosaic	Cyberpunk	Line Drawing
Impressionism	Cubism	Surrealism
Pop Art	Minimalism	Abstract Expressionism
Pointillism	Graffiti	Neo-Futurism
Baroque	Melting Gold	Melting Silver
Gold Material	Made by Gold	Made by Bronze
Fantasy Glow Rendering	Low-Poly	

Table 12. **Style key words** for FLUX to generate stylised images.

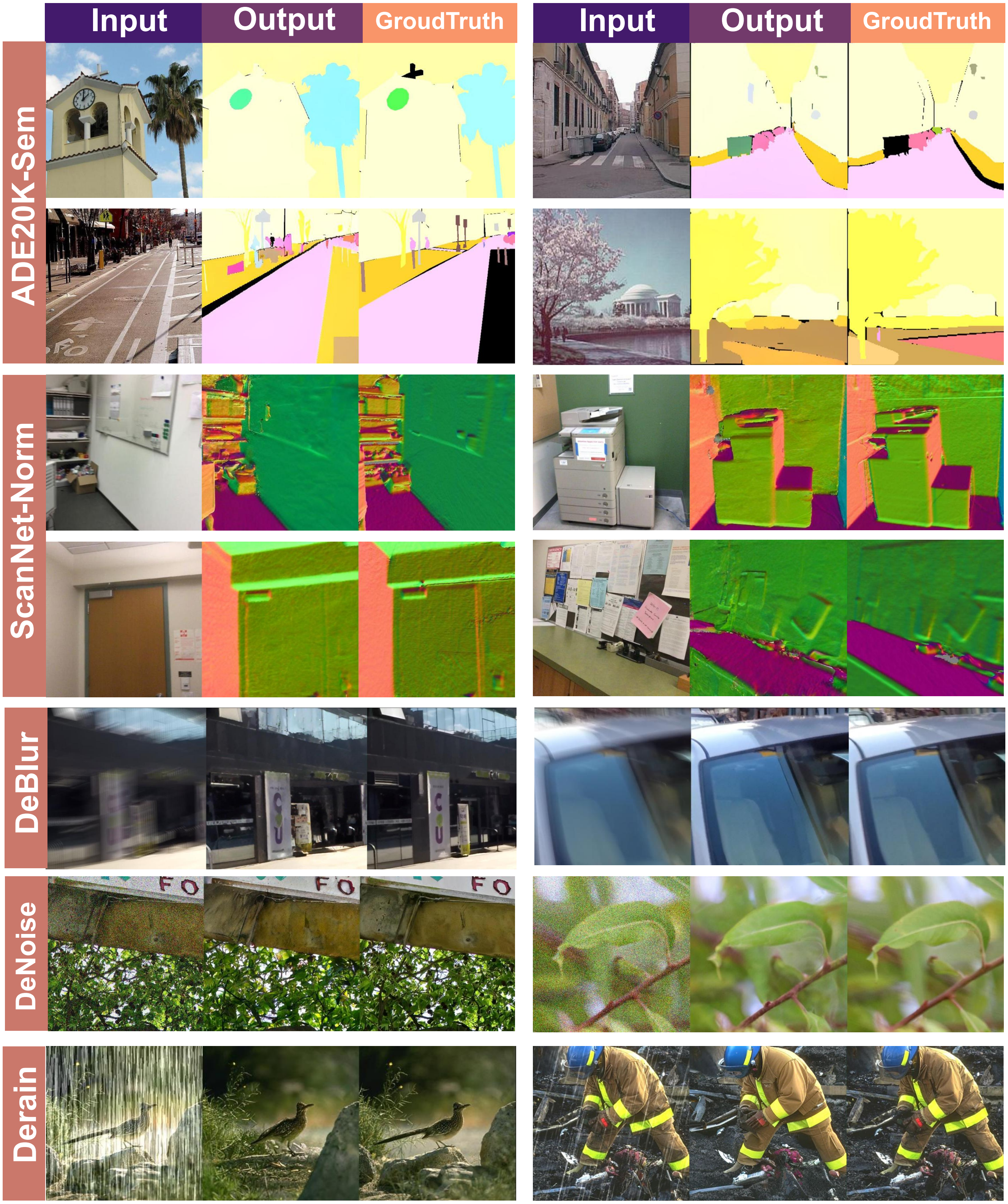
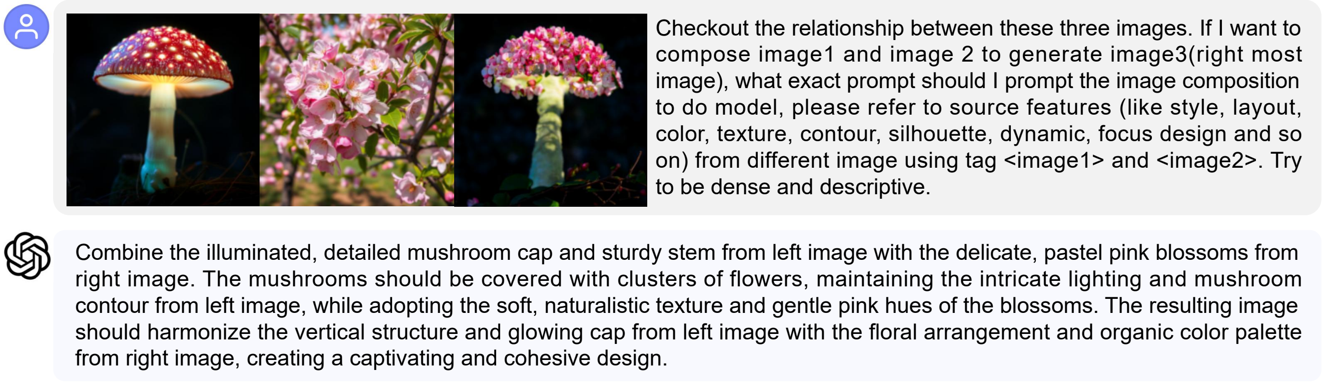


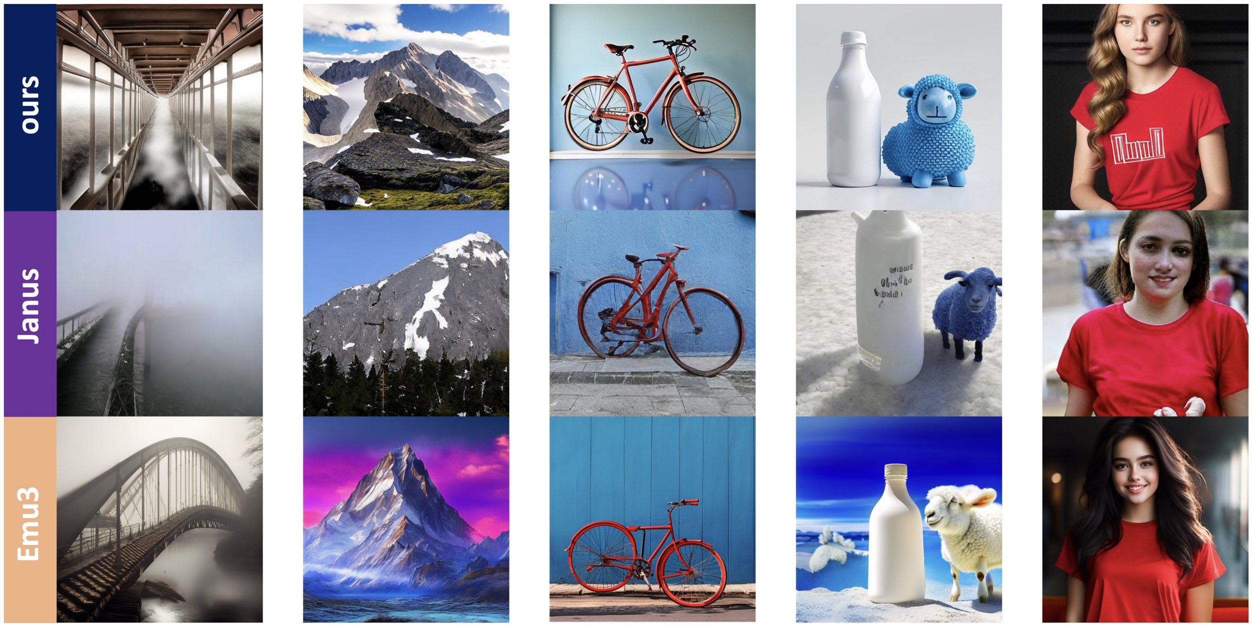
Figure 9. **Qualitative results of diversed tasks,** such as semantic segmentation, norm estimation, image deblur, denoise and derain.



Checkout the relationship between these three images. If I want to compose image1 and image 2 to generate image3(right most image), what exact prompt should I prompt the image composition to do model, please refer to source features (like style, layout, color, texture, contour, silhouette, dynamic, focus design and so on) from different image using tag <image1> and <image2>. Try to be dense and descriptive.

Combine the illuminated, detailed mushroom cap and sturdy stem from left image with the delicate, pastel pink blossoms from right image. The mushrooms should be covered with clusters of flowers, maintaining the intricate lighting and mushroom contour from left image, while adopting the soft, naturalistic texture and gentle pink hues of the blossoms. The resulting image should harmonize the vertical structure and glowing cap from left image with the floral arrangement and organic color palette from right image, creating a captivating and cohesive design.

Figure 10. **An example of conversation with GPT-4o** to annotate the relationship between input images and output image produced by IP-Adapter [79]



A bridge with no end vanishing into the fog

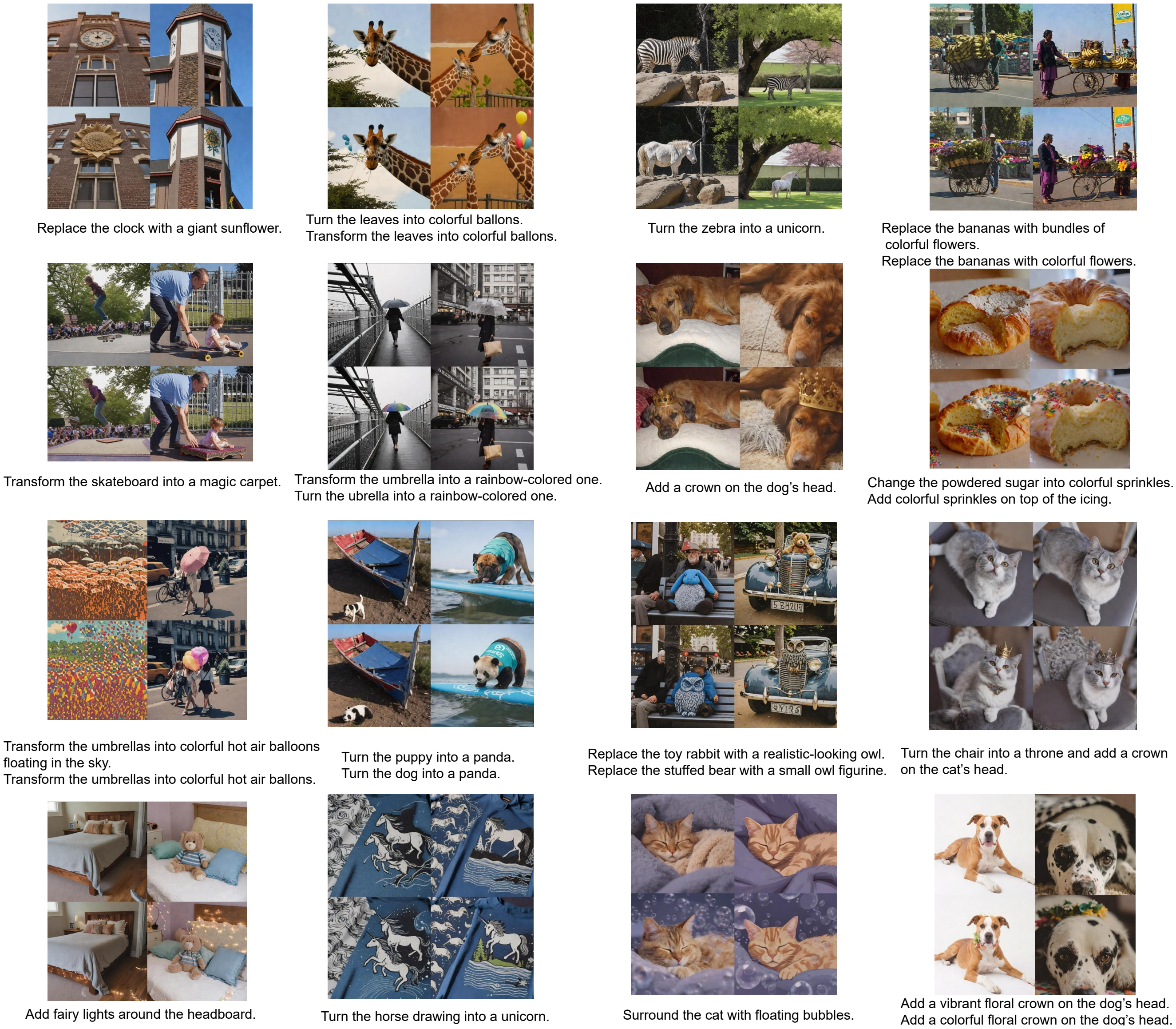
A mountain with no snow on its peak

A red bicycle against a blue wall

A white bottle and a blue sheep

A young woman wearing a red T-shirt

Figure 11. **Qualitative results of text-to-image generation.** Compared to Janus [71] and Emu3 [69], our model presents marked improvement in both quality and textual alignment.



Replace the clock with a giant sunflower.

Turn the leaves into colorful ballons.
Transform the leaves into colorful ballons.

Turn the zebra into a unicorn.

Replace the bananas with bundles of colorful flowers.
Replace the bananas with colorful flowers.

Transform the skateboard into a magic carpet.

Transform the umbrella into a rainbow-colored one.
Turn the ubrella into a rainbow-colored one.

Add a crown on the dog's head.

Change the powdered sugar into colorful sprinkles.
Add colorful sprinkles on top of the icing.

Transform the umbrellas into colorful hot air balloons floating in the sky.
Transform the umbrellas into colorful hot air ballons.

Turn the puppy into a panda.
Turn the dog into a panda.

Replace the toy rabbit with a realistic-looking owl.
Replace the stuffed bear with a small owl figurine.

Turn the chair into a throne and add a crown on the cat's head.

Add fairy lights around the headboard.

Turn the horse drawing into a unicorn.

Surround the cat with floating bubbles.

Add a vibrant floral crown on the dog's head.
Add a colorful floral crown on the dog's head.

Figure 12. **Visualization of in-context training example in RAIE.** After CLIP [44] based clustering, many instruction are similar or completely the same, which is crucial to the success of RAIE.