A. Implementation Details

A.1. Method Configuration

Our implementation builds upon the official codebases of LlamaGen [11] and Lumina-mGPT [7]. For LlamaGen-based editing, we employ a candidate window size of $K = 150$ and similarity threshold $\tau = 1.0$, utilizing Euclidean distance metrics in the VQ-AutoEncoder codebook's latent space. The Lumina-mGPT implementation adopts $K = 100$ and $\tau = 0.4$, measuring token distances through cosine similarity of first-layer transformer embeddings. All experiments were conducted on NVIDIA 3090 GPUs.

A.2. Baseline Implementation

For the diffusion-based methods we compared, including P2P [5], Null-text inversion [8], PnPInversion [6], Pix2Pix-zero [9], MasaCtrl [3], InstructPix2Pix [2], and MGIE [4], we utilized their official implementations. For the two simple autoregressive (AR) model-based baselines we implemented:

- **Naive Modify Prompt (*NPM*)**: This baseline modifies the original prompt to the edited prompt while keeping all other variables (e.g., non-edited words and random seeds) unchanged.
- **PnP-AR**: In this baseline, we save the token-wise and layer wise attention maps computed during the generation process of the original prompt. These attention maps are then directly replaced at the corresponding token positions and layers when generating images from the edited prompt.

A.3. Evaluation Benchmarks

Our method focuses on five fundamental editing types: object replacement, object addition, object deletion, style transfer, and attribute modification. For each editing type, we randomly select 10 examples from the corresponding category in the PIE-Bench [6] dataset. Each example includes an original prompt and an edited prompt.

We first use LlamaGen to generate images based on the original prompts and then apply our method to edit these images according to the edited prompts. Due to the inherent limitations of LlamaGen in generating high-quality results from short prompts [11], we employ GPT-4o mini [1] as a prompt enhancer to refine and improve the prompts before generation.

B. Attention Map Analysis

B.1. Attention mechanism in AR

As illustrated in Figure 1, for autoregressive image generation models such as LlamaGen, the text prompt is first encoded by the text encoder to obtain text tokens, which serve as the prefix tokens for the entire generation sequence. During the generation of each subsequent image token, attention is computed with both the preceding image tokens and the entire set of text tokens. The self-attention maps presented in our paper are derived from the image self-attention mechanism, as shown in the bottom-right section of Figure 1, while the cross-attention maps are obtained from the image-to-text cross-attention mechanism, depicted in the bottom-left section of Figure 1.

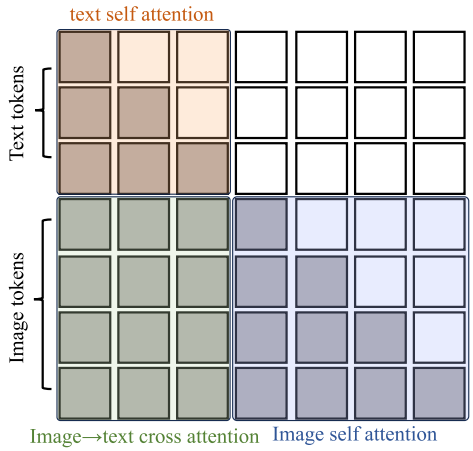

Figure 1. Illustration of the Attention mechanism in LlamaGen [10].

B.2. Attention Visualization

Our method does not explicitly inject attention maps. Instead, structural preservation is implicitly achieved through anchor token matching, which naturally results in attention map consistency as a byproduct. As shown in Figure 2, compared to the attention maps obtained using the NPM method, the attention maps of the edited images generated by our method align naturally with those of the original images.

B.3. Attention Locality

We observe that in autoregressive models such as LlamaGen, tokens tend to allocate higher attention weights to those adjacent to their positions during attention computation. As shown in Figure 3, the attention score assigned by the current token decreases as the distance from the current token increases. However, the attention score periodically increases at intervals of 32 tokens. This phenomenon occurs because, when generating 512×512 images, LlamaGen employs a VQ-Autoencoder to encode the image into a 32×32 latent space. Tokens located at multiples of 32 positions away from the current token reside in the same column in the latent space, resulting in higher attention scores at these intervals. This also explains why the cross-attention from image tokens to text tokens in Figure 4 of main paper shows that the earliest image tokens have the highest attention scores.

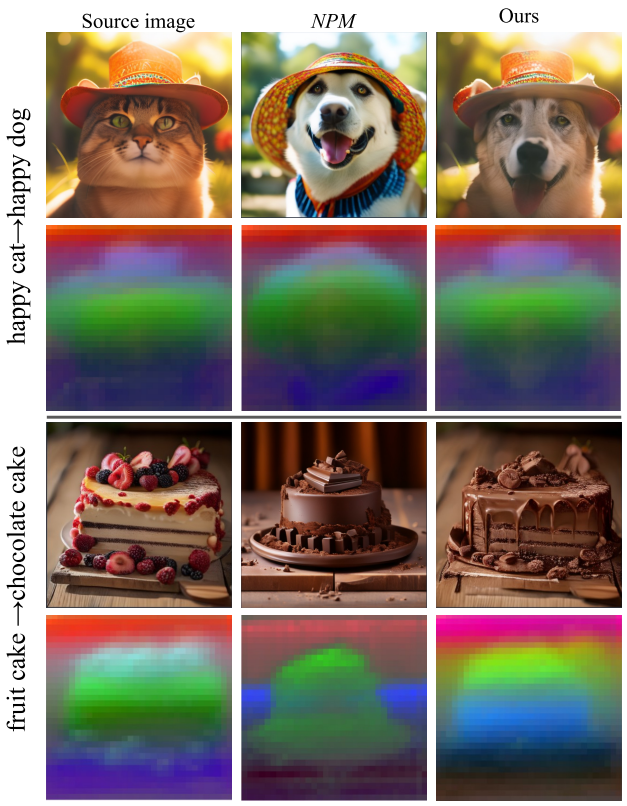

Figure 2. Ablation studies on attention maps. Compared with *NPM*, our method *ISLock* naturally achieves better alignment of the original image and edited image generation processes during the attention map process.

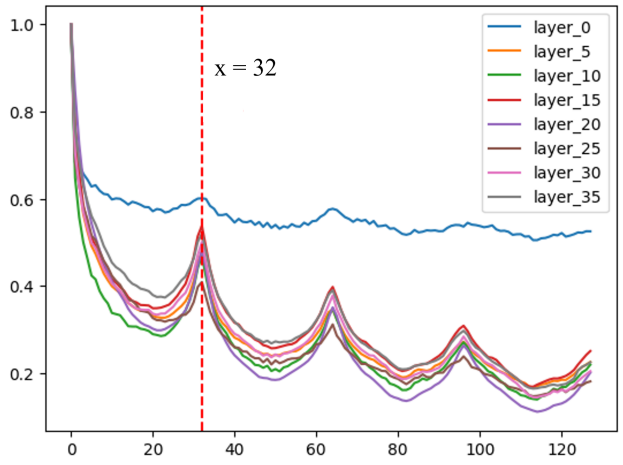

Figure 3. Autoregressive image generation models often tend to allocate larger attention score to tokens at adjacent positions.The values in the figure are normalized using min-max normalization.

C. More Visualization Comparisons

In Figure 4, we present additional editing results, demonstrating that our method generalizes well across different editing types and AR-based models.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 1

[3] 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. *ICCV*, 2023. 1

[4] Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. *arXiv preprint arXiv:2309.17102*, 2023. 1

[5] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *ICLR*, 2023. 1

[6] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct inversion: Boosting diffusion-based editing with 3 lines of code. *ICLR*, 2023. 1

[7] Dongyang Liu, Shitian Zhao, Le Zhuo, Weifeng Lin, Yu Qiao, Hongsheng Li, and Peng Gao. Lumina-mgpt: Illuminate flexible photorealistic text-to-image generation with multimodal generative pretraining. *arXiv preprint arXiv:2408.02657*, 2024. 1, 3

[8] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *CVPR*, 2023. 1

[9] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 2023. 1

[10] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. LLama-Gen:Autoregressive Model Beats Diffusion: Llama for Scalable Image Generation. 2024. 1, 3

[11] 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. 1

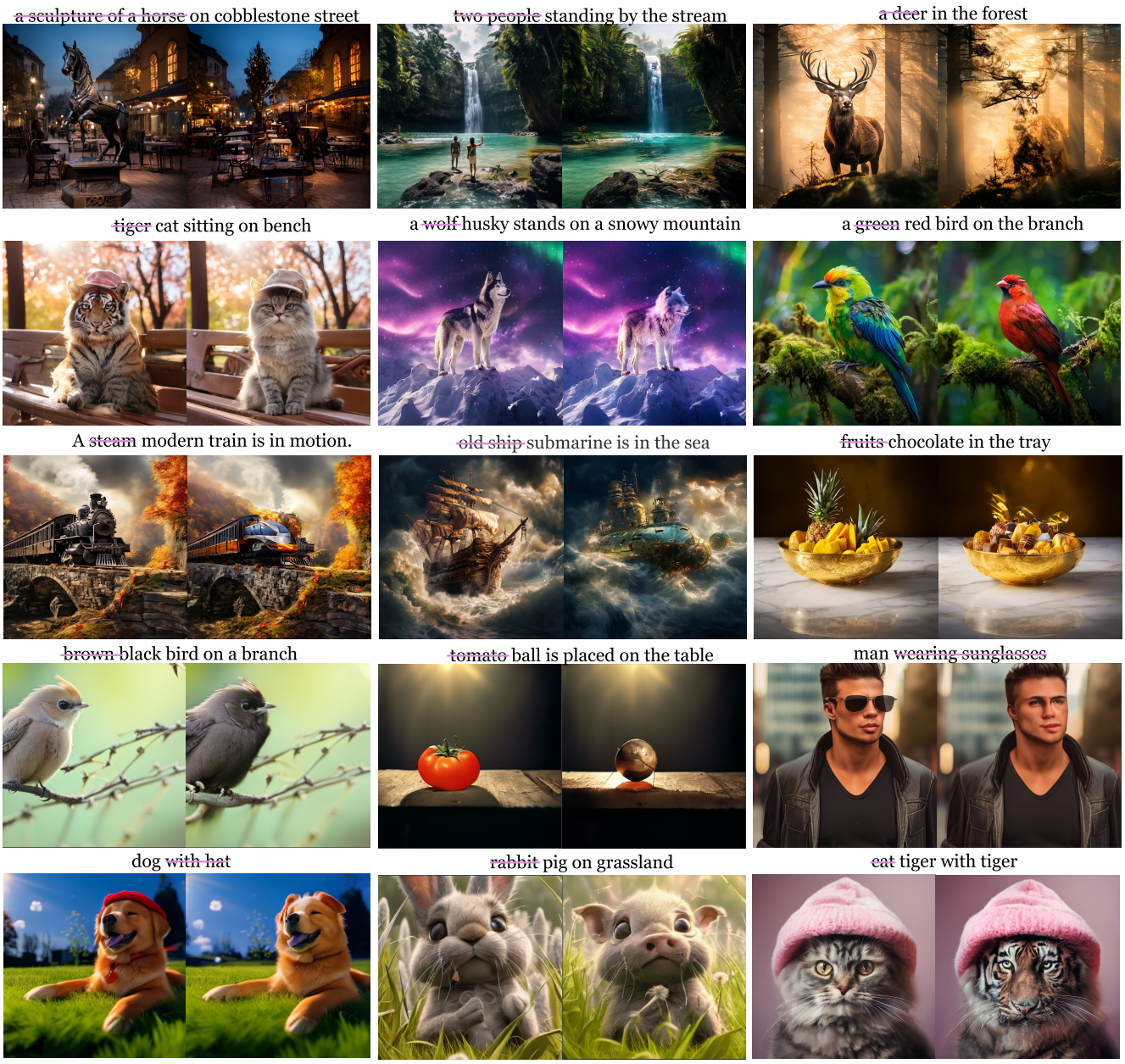

Figure 4. More visualization results of our method *ISLock*, where in the first three rows our method is integrated with lumina-mgpt [7] and in the last two rows it is working with LlamaGen [10].