Exploring Multimodal Diffusion Transformers for Enhanced Prompt-based Image Editing

Supplementary Material

A. Editing Real Images

A.1. Rectified flows

All multimodal diffusion transformers (MM-DiT) models discussed in our paper use the setting of Rectified flows [35] for noise scheduling and sampling. Rectified flow presents an approach to learn ordinary differential equation (ODE) for transporting between two distributions π_0 and π_1(image distribution and standard Gaussian, respectively). The key idea is to learn an ODE that follows straight paths connecting points drawn from $X_0 \sim \pi_0$ and $X_1 \sim \pi_1$ as closely as possible, formulated as follows.

$$X_t = (1-t)X_0 + tX_1, \quad t \in [0,1] \tag{A1}$$

$$dZ_t = v_t(Z_t)dt, \tag{A2}$$

$$v_t(x) = \mathbb{E}\left[\dot{X}_t \middle| X_t = x\right] = \mathbb{E}\left[X_1 - X_0 | X_t = x\right], \tag{A3}$$

$\dot{X}_t$ denotes the time differential of X_t. Eq. (A1) defines the marginal density at time t, corresponding to noise scheduling in diffusion models. Eq. (A2) and Eq. (A3) explain the flow connecting each sample $Z_0 \leftarrow \pi_0$ and $Z_1 \leftarrow \pi_1$.

$v_t(x;\phi)$ is optimized and evaluated using a neural network through the tractable conditional flow matching objective, where ϕ represents the trainable parameters of the model.

$$\mathcal{L}_{\text{CFM}}(\phi) := \mathbb{E}_{t,X_t,X_1}\left[\|v_t(X_t|X_1) - v_t(X_t;\phi)\|_2^2\right], \\ \text{where } t \sim \mathcal{U}[0,1], X_t \sim p_t(\cdot|X_1), X_1 \sim \pi_1. \tag{A4}$$

$$v_t(x|X_1) = \frac{X_1 - x}{1-t}, \quad v_t(x|X_0) = \frac{x - X_0}{t}, \tag{A5}$$

Eq. (A5), derived from Eq. (A1) and Eq. (A3), shows that the conditional flow follows a straight line to its destination.

A.2. RF inversion

Rout et al. [56] proposed a novel inversion framework for RF models that address inversion and editing tasks. Inversion is achieved by following a controlled forward ordinary differential equation (ODE), establishing a mapping from the real image distribution π_0 to the standard Gaussian distribution π_1 for the recovery of a noisy latent representation from a given real image. Conversely, the controlled reverse ODE enables editing by starting with a sample from π_1 and mapping it back to π_0, where additional guidance can be applied using target prompts.

The *controlled forward ODE* maps real image samples from π_0 to standard Gaussian π_1 as t progresses from 0 to 1. Let $\mathbf{x}_1 \leftarrow \pi_1$ denote a sample from the standard Gaussian distribution, which serves as a regulation point for inversion. The controlled forward vector field $\hat{v}_t$ is defined as:

$$\hat{v}_t(X_t) = v_t(X_t) + \gamma\big(v_t(X_t \mid \mathbf{x}_1) - v_t(X_t)\big), \quad t \in [0,1]. \tag{A6}$$

$$v_t(X_t) = v\big(X_t, t, \Phi(``"); \phi\big), \tag{A7}$$

$$v_t(X_t \mid \mathbf{x}_1) = \frac{\mathbf{x}_1 - X_t}{1-t}, \tag{A8}$$

The controlled vector field $\hat{v}_t$ is constructed as a weighted interpolation between the unconditional vector field $v_t(\cdot)$ with the null prompt guidance through text encoder Φ, and the conditional vector field $v_t(\cdot \mid \mathbf{x}_1)$, which guides the latent variable toward $\mathbf{x}_1$ to align better the target distribution π_1. Hyperparameter γ controls the degree of interpolation between these two fields.

The *controlled reverse ODE* maps a sample from π_1 back to π_0, with t progressing from 1 to 0, effectively reversing the forward process. Solving this ODE enables reconstruction and editing, with the latter guided by a target prompt. The controlled vector field $\hat{v}_t$ for reverse transformation is defined as:

$$\hat{v}_t(X_t) = v_t(X_t) + \eta\big(v_t(X_t \mid \mathbf{x}_0) - v_t(X_t)\big), \quad t \in [0,1]. \tag{A9}$$

$$v_t(X_t) = v\big(X_t, t, \Phi(\text{target prompt}); \phi\big), \tag{A10}$$

$$v_t(X_t \mid \mathbf{x}_0) = \frac{X_t - \mathbf{x}_0}{t}, \tag{A11}$$

The formulation is similar to Eq. (A6), expressed as a weighted combination of the unconditional vector field $v_t(\cdot)$ with target prompt guidance, and the conditional vector field $v_t(\cdot \mid \mathbf{x}_0)$, which incorporates the reference image $\mathbf{x}_0$ to align the output with the original real image. The interpolation between these fields is governed by the hyperparameter η. Please refer to [56] for a detailed theoretical derivation.

A.3. Qualitative comparison with other methods

As discussed in Sec. 5, our method can be effectively combined with inversion techniques. We first obtain the initial latent through inversion or sampling from a Gaussian distribution. We then define a conditional interpolation path between this initial latent and the image latent using Eq. (A1), which we treat as the source branch. During denoising, we simultaneously evaluate the model using both the source and target branches, replacing the target branch's input projections with those derived from the source branch. Note that the source branch outputs are solely employed to obtain the $\mathbf{q}_i$ and $\mathbf{k}_i$ projections required to update the target branch. The remaining outputs from the source branch are disregarded, as the interpolation path between the initial latent and source image latent is already theoretically defined.

We provide qualitative comparisons in Fig. A1, against several baseline methods: (1) SDEdit [41] based on SD1, (2) SDEdit based on Flux.1-dev, (3) Null-text inversion (NTI) [42] with Prompt-to-Prompt [19] based on SD1, and (4) RF inversion [56], along with our results both with and without RF inversion. For implementation, we used community implementations for SDEdit variants and RF inversion and the official implementation for NTI+P2P. SD1-based methods were experimented with default settings (50 timesteps at 512×512 resolution), while Flux.1-dev-based methods used 28 timesteps and 1024×1024 resolution. As mentioned in Sec. 4.1, original P2P only allows changes with the same word counts, and it does not support changing words like 'Cappadocia' to 'Niagara Falls' due to different word counts. To address this limitation, we manually modified some prompts for NTI+P2P by removing spaces between words (*i.e.*, 'NiagaraFalls'). All baseline hyperparameters were empirically optimized.

In general, SD1-based methods occasionally show limitations in output quality due to the base model's capacity. Compared to SDEdit (Flux.1) and RF inversion, our method enables larger changes while naturally preserving unmodified regions. When our approach is applied without inversion, results become more sensitive to the local blending threshold (θ), requiring higher thresholds to effectively maintain targeted regions due to divergence in the denoising sequence. In contrast, starting from optimized inverted latents inherently preserves source image characteristics, making results less sensitive to threshold values, as lower thresholds are already sufficient.

A.4. Quantitative comparison with other methods

We evaluate our method on PIE-Bench [27], a benchmark for prompt-based image editing consisting of 700 samples. To maximize the model's performance, we compare Flux.1 and SD 1.4-based methods at their native resolutions of 1024 and 512, respectively. Our inversion-free method requires careful local blending threshold θ control, so we only perform experiments with inverted latents using RF inversion and fixed blending threshold. As seen in Tab. A1, our approach improves upon RF inversion with increased controllability via θ. While lowering θ enables broader edits, selecting an appropriate θ allows our method to improve RF inversion in both edit quality and image preservation.

Overall, we observed that Flux-based methods effectively reflect the desired edit prompts but demonstrate relatively weaker identity preservation than SD 1-based methods. We identify two main reasons behind this:

1) Inversion method: RF inversion employs first-order Euler methods using a controlled vector field derived through dynamic optimal control, interpolating between two vector fields: an unconditional vector field guiding images to noise, and a controlled vector field that ensures the inverted latent to be closer to a "typical" latent of Flux's latent distribution. While RF inversion performs reasonably well, starting from better inverted latents could potentially yield higher scores. As can be seen with the SD1 case, changing the inversion method from first-order DDIM to more advanced PnP inversion improves scores throughout. Additionally, RF inversion utilizes a controlled interpolation mechanism explicitly designed to guide inverted latents toward Flux's distribution of clean images, sometimes producing reconstruction that appear perceptually sharper or cleaner than the source images, paradoxically leading to lower reconstruction metrics such as PSNR and LPIPS.

2) Dataset characteristics: PIE-Bench, proposed by [27], was mostly developed within SD1's capabilities, involving relatively simpler and slightly noisy images at SD1's favorable 512×512 resolution. When running at Flux.1's native 1024×1024 resolution, the refinement effect often adversely affects identity preservation metrics. Additionally, the benchmark contains relatively fewer examples that can properly evaluate MM-DiT's complex and precise control capabilities, such as editing text.

PnP [27] achieved strong performance through extensive grid searches for optimal hyperparameters. In contrast, RF inversion performs reasonably well using only first-order Euler steps and controlled interpolation, without thorough hyperparameter tuning. Our visual inspection on edited images revealed many high-quality results with RF inversion, and our method enhances them with additional controllability. As existing benchmarks tend to use simpler scenes (mostly fitted to SD1), we believe evaluating larger models on more complex scenes and tasks (*e.g.*, text rendering, or high-resolution broader edits) remains an important direction.

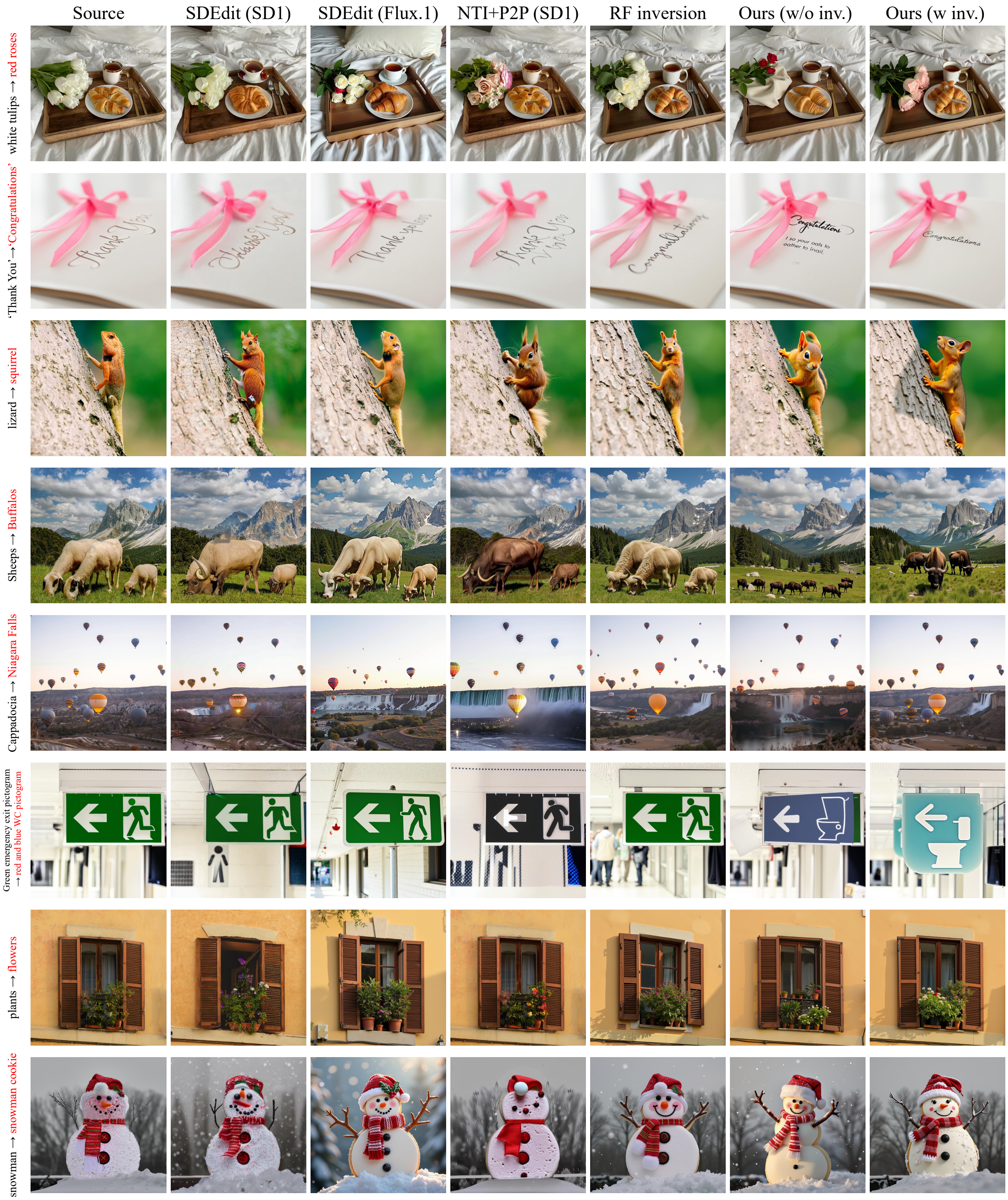


Figure A1. Qualitative comparison of real image editing methods. We evaluate using diverse real images from Pexels and Pixabay, with their initial captions generated by LLM [45] and subsequently modified for editing tasks. Best viewed zoomed in.

Table A1. Comparison of diverse image editing methods in PIE-Bench [27]. Best results are in **bold** and second best are underlined, ranked separately for SD 1.4 and Flux.1-dev methods.

Method	Model / Steps	Structure	Background Preservation				CLIP Similarity	
		Distance ↓	PSNR ↑	LPIPS ↓	MSE ↓	SSIM ↑	Whole ↑	Edited ↑
InstructPix2Pix [3]	SD 1.4 / 50 steps	0.057	20.85	0.158	0.0227	0.768	23.90	21.74
InstructDiffusion [17]	SD 1.4 / 50 steps	0.075	20.31	0.155	0.0349	0.761	23.46	21.38
P2P (DDIM-Inv) [19]	SD 1.4 / 50 steps	0.070	17.88	0.208	0.0219	0.717	25.31	22.57
Pix2PixZero (DDIM-Inv) [46]	SD 1.4 / 50 steps	0.062	20.46	0.172	0.0144	0.753	23.07	20.64
MasaCtrl (DDIM-Inv) [4]	SD 1.4 / 50 steps	0.027	22.19	0.106	0.0087	0.803	24.23	21.25
P2P (PnP-Inv) [27]	SD 1.4 / 50 steps	**0.011**	**27.28**	**0.054**	**0.0032**	**0.853**	**25.34**	**22.17**
Pix2PixZero (PnP-Inv) [27]	SD 1.4 / 50 steps	0.050	21.56	0.138	0.0127	0.777	23.64	21.15
MasaCtrl (PnP-Inv) [27]	SD 1.4 / 50 steps	0.024	22.66	0.087	0.0081	0.819	24.70	21.45
RF inversion [56]	Flux.1-dev / 28 steps	0.026	23.73	0.144	0.0065	0.769	24.56	21.59
Ours (θ=0.2, RF-inv)	Flux.1-dev / 28 steps	0.054	19.92	0.204	0.0174	0.731	**25.43**	**22.56**
Ours (θ=0.5, RF-inv)	Flux.1-dev / 28 steps	**0.025**	**24.79**	**0.126**	**0.0059**	**0.804**	24.62	21.61

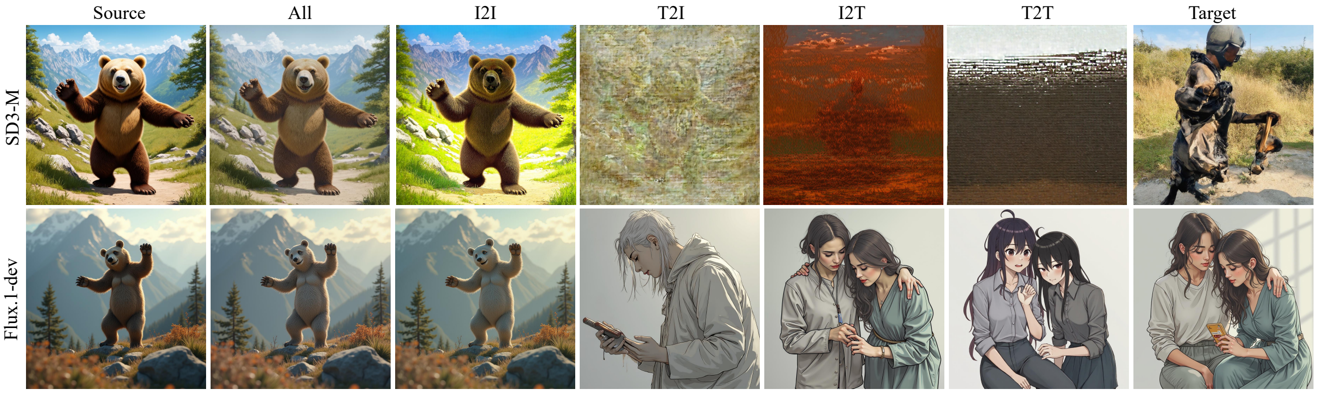


Figure A2. Analyzing attention map components by replacing different portions from a source prompt ("a photo-realistic bear dancing in the mountain") to an empty target prompt (""). Results show I2I portions primarily preserve spatial layout and geometry, with T2T adding the most negligible impact. Full attention map replacement produces the closest match to source image.

B. Block-wise Attention Patterns

B.1. Additional discussions on I2I & T2T blocks

As mentioned in the main paper, the I2I block is analogous to self-attention in U-Net architectures, effectively capturing spatial layout and geometric information. In contrast, T2T blocks primarily manifest as identity matrices, indicating strong self-correlation among tokens. To validate the relative importance of these sub-blocks, we conducted experiments injecting attention maps from meaningful prompts into the empty prompt ("") branches (Fig. A2). While full attention map transfer produced the closest replication of source images, we found that the I2I block alone sufficiently preserves geometric structure, whereas T2T has minimal impact.

To further investigate T2T blocks, we visualize their attention patterns using the prompt "a panda riding a bicycle on the beach under blue sky" in Fig. A18. For the SD3-M variant, which utilizes 333 tokens (77 CLIP + 256 T5), we observe pronounced attention signals around special tokens, particularly at sequence boundaries such as start/end tokens and transitions from CLIP to T5 embeddings. Similarly, Flux.1-dev, which employs 512 T5 tokens exclusively, also exhibits notable attention at prompt endings, with attention weights substantially decreasing after meaningful tokens (*e.g.*, EOS). These patterns suggest a focused allocation of attention toward semantically relevant token boundaries. Additional subtle and noisy patterns within T2T blocks require further exploration, which we defer to future work due to T2T blocks' minimal impact on current editing scenarios.

B.2. Additional discussions on T2I & I2T blocks

We begin by visualizing the T2I and I2T portions of attention maps across several model variants (SD3-M: Fig. A14, SD3.5-M: Fig. A15, SD3.5-L: Fig. A16, Flux.1: Fig. A17). As discussed in the main paper, we observe spatially and

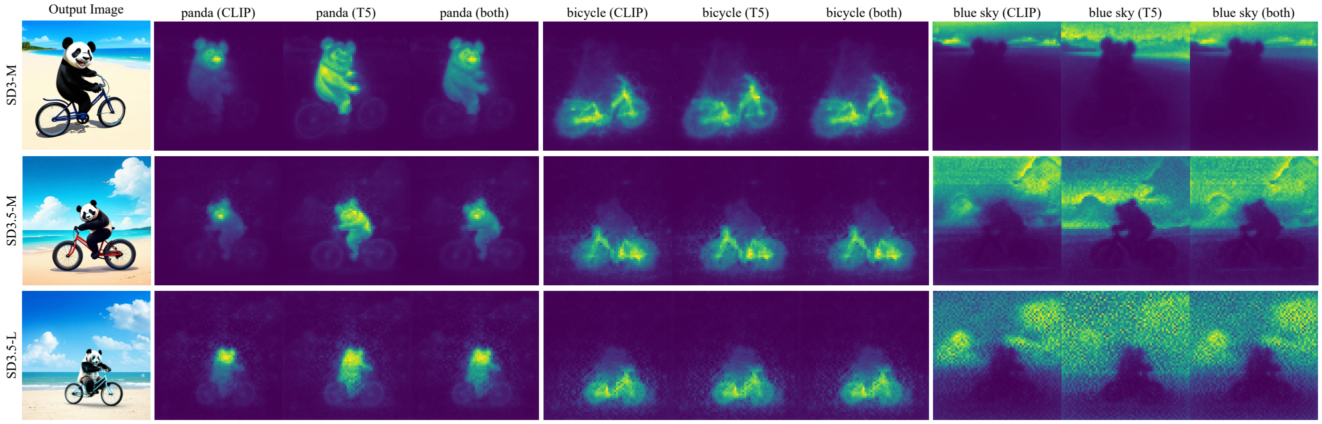


Figure A3. Visualization of T2I attention maps separated by text encoder type. For SD3 variants using both CLIP and T5 embeddings, we visualize attention patterns from CLIP tokens, T5 tokens, and their combination by tracking respective token positions.

geometrically aligned visual patterns when we visualize attention patterns between image tokens and specific text tokens. This alignment indicates that each domain preserves its distinct characteristics even within the multimodal full attention mechanism. Notably, this phenomenon persists in Flux.1's single-branch blocks, where a unified set of weights processes concatenated tokens. This observation suggests that the architectural choice between dual and single branches does not largely compromise the model's ability to maintain domain-specific features. It is also worth noting that certain blocks produce extremely noisy attention maps, which validates our strategy of utilizing only selected well-defined blocks for local blending to achieve more precise local edits.

B.3. Comparing CLIP and T5 text encoders

Another notable aspect is the use of T5 text encoders alongside CLIP text encoders in SD3 series. As shown in Tab. 1, all Stable Diffusion 3 series models we tested (SD3-M, SD3.5-M, SD3.5-L) utilize three text encoders for the text branch in MM-DiT, concatenating two CLIP text embeddings with T5 text embeddings along the sequence dimension, whereas Flux.1 exclusively uses T5 for the text branch and utilizes CLIP features only as pooled embeddings for scale and shift operations. In Fig. A3, we visualize CLIP and T5 attention patterns separately. CLIP text encoders generally produce denser, more localized attention patterns focused on specific regions. In contrast, T5 encoder generates more spread-out attention patterns that appear more contextual, sometimes extending to related concepts (*e.g.*, "blue sky" attention spreading to ocean regions due to shared blue attributes). In SD3 / 3.5 architectures, we naturally utilize attention maps from both CLIP and T5 text token positions when aggregating T2I blocks to generate local blending masks.

B.4. Detailed explanation of token misalignment

In Sec. 4.1, we discussed how changing the entire attention map can lead to misalignment with the value matrix. Here, we explain this using the example prompt "a panda riding a bicycle on the beach under blue sky". The CLIP tokenizer produces ['a', 'panda', 'riding', 'a', 'bicycle', 'on', 'the', 'beach', 'under', 'blue', 'sky'], while the T5 tokenizer yields ['', 'a', 'pan', 'd', 'a', 'riding', '', 'a', 'bicycle', 'on', 'the', 'beach', 'under', 'blue', 'sky']. When editing with a similar prompt "a dragon riding a bicycle on the beach under blue sky", CLIP tokenization simply requires mapping 'panda' to 'dragon' as they are both single tokens. However, in T5, we need to create a mapper that maps all three tokens ('pan', 'd', 'a') to a single 'dragon' token. While P2P handles such cases by defining explicit token mappings, this approach becomes challenging with drastically different prompts like "a princess with a crown riding an elephant on the beach under blue sky", where determining appropriate token correspondences is non-trivial. This limitation becomes more pronounced in larger models with longer, more descriptive prompts and T5 tokenization. As shown in Fig. 10, while naive attention map replacement leads to undesired changes due to these token misalignments, our approach of modifying only image tokens naturally circumvents this limitation by keeping text token projections intact.

C. In-depth Analysis of Transformer Blocks

C.1. Identifying effective transformer blocks for obtaining clearer attention maps

In Sec. 3.3, we presented our analysis of transformer blocks in Flux.1 to identify those producing clear attention maps suitable for local blending. Here, we extend this analysis to additional model architectures: SD3-M (Fig. A4), SD3.5-

Table A2. Top-5 block indices calculated using rankings of BCE loss, Soft mIoU, and MSE. Results shown with and without Gaussian smoothing across different model variants.

Model	w/o Gaussian Smoothing	w. Gaussian Smoothing
SD3-M	[7, 8, 5, 4, 9]	[7, 8, 5, 4, 9]
SD3.5-M	[7, 8, 5, 9, 6]	[7, 9, 8, 5, 10]
SD3.5-L	[18, 16, 29, 21, 14]	[18, 21, 20, 24, 16]
Flux.1-dev	[11, 50, 18, 13, 10]	[18, 17, 12, 14, 11]

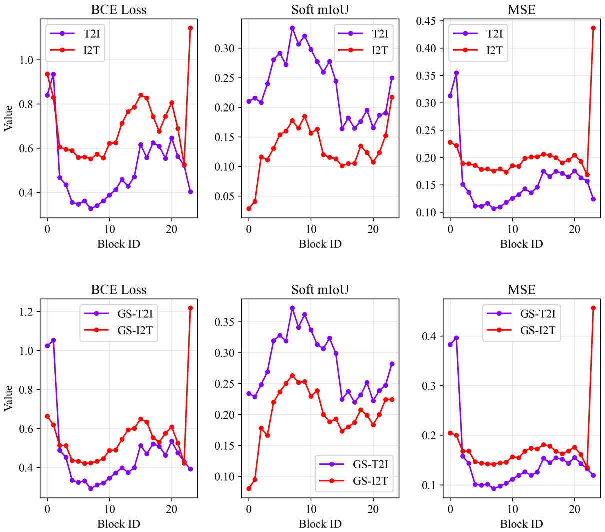


Figure A4. Transformer block analysis of SD3-M using Binary Cross Entropy Loss, Soft mIoU, and MSE, with Grounded SAM2 predictions as ground truth. Scores are shown without (upper) and with (lower) Gaussian smoothing.

M (Fig. A5), and SD3.5-L (Fig. A6). We evaluate transformer blocks for each architecture using three metrics - Binary Cross Entropy Loss, Soft mIoU, and MSE - both with and without Gaussian smoothing. The top-5 blocks selected based on these metrics are summarized in Tab. A2.

As mentioned in the main paper, smaller models (SD3-M, SD3.5-M) largely maintain their block rankings regardless of Gaussian smoothing application. In contrast, larger models (SD3.5-L and Flux.1-dev) show significant changes in block rankings after smoothing. This suggests that while some blocks in larger models appear noisy in their raw form, they contain valuable structural information that becomes apparent after smoothing.

We utilize the T2I portions of the identified top-5 transformer blocks for local blending operations. We selectively compute full attention maps only for these 5 blocks while using PyTorch's optimized SDPA kernel for all other blocks. This approach achieves both precise attention control and computational efficiency. The resulting attention maps from these selected blocks serve as the foundation for our local blending mechanism, enabling precise and controlled image editing. Qualitative results with and without local blending are shown in Fig. A7 for comparison.

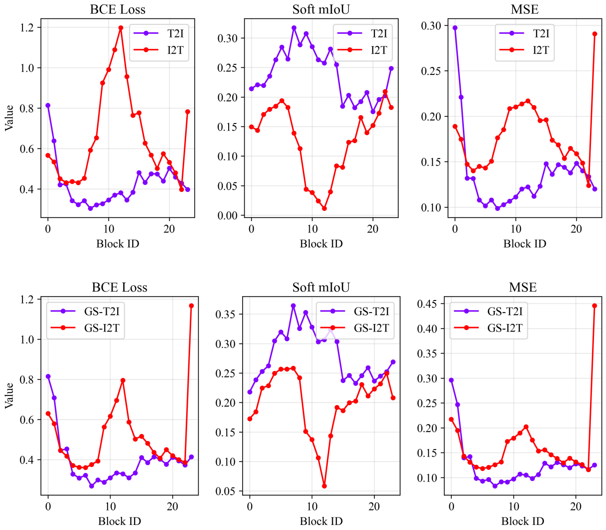


Figure A5. Transformer block analysis of SD3.5-M using Binary Cross Entropy Loss, Soft mIoU, and MSE, with Grounded SAM2 predictions as ground truth. Scores are shown without (upper) and with (lower) Gaussian smoothing.

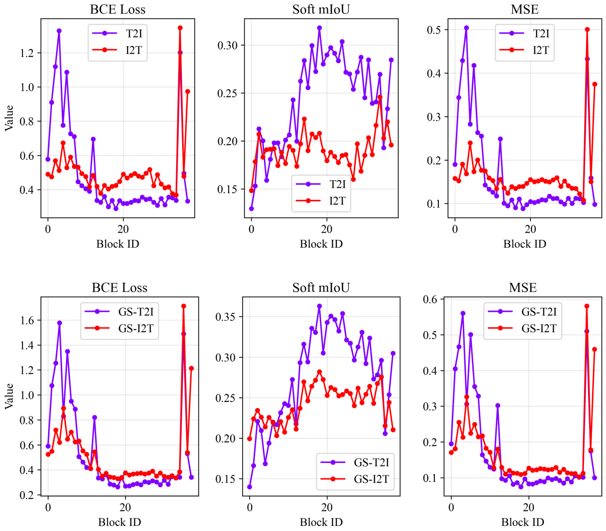


Figure A6. Transformer block analysis of SD3.5-L using Binary Cross Entropy Loss, Soft mIoU, and MSE, with Grounded SAM2 predictions as ground truth. Scores are shown without (upper) and with (lower) Gaussian smoothing.

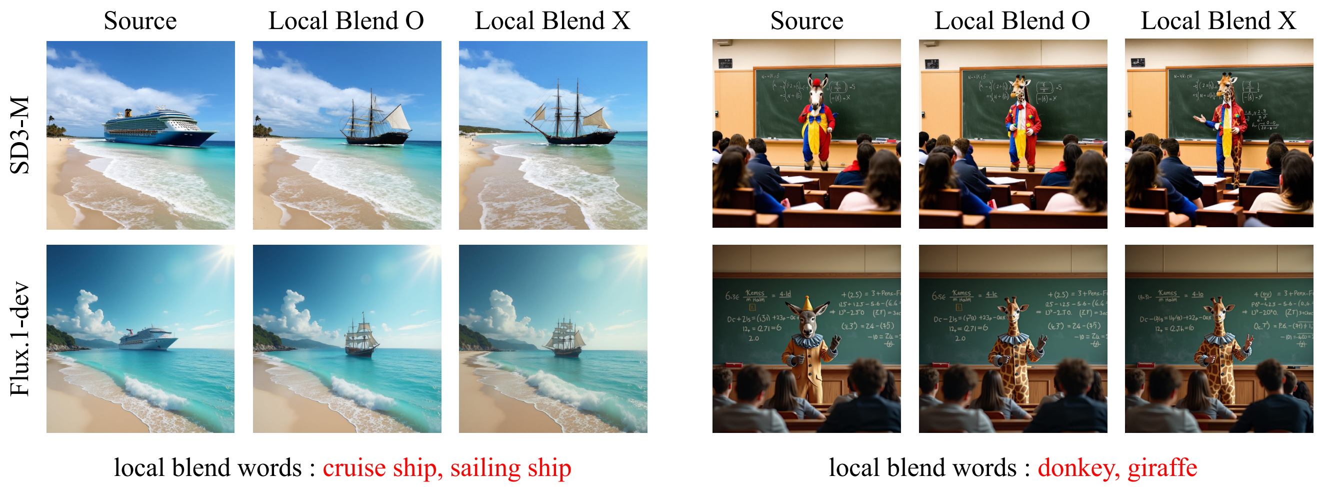

Source
Local Blend O
Local Blend X
Source
Local Blend O
Local Blend X
SD3-M
Flux.1-dev
local blend words : cruise ship, sailing ship
local blend words : donkey, giraffe

Figure A7. Local blending effects in SD3-M and Flux.1-dev models. The method excels at preserving non-targeted elements: in the maritime scene, the shoreline, island formations, and cloud patterns remain unchanged; in the classroom scene, the blackboard content and student arrangements are preserved. Here, we used the previously identified top-5 blocks from each model to generate masks with a local blending threshold of 0.4, applying the blending up to 50% of total timestep iterations.

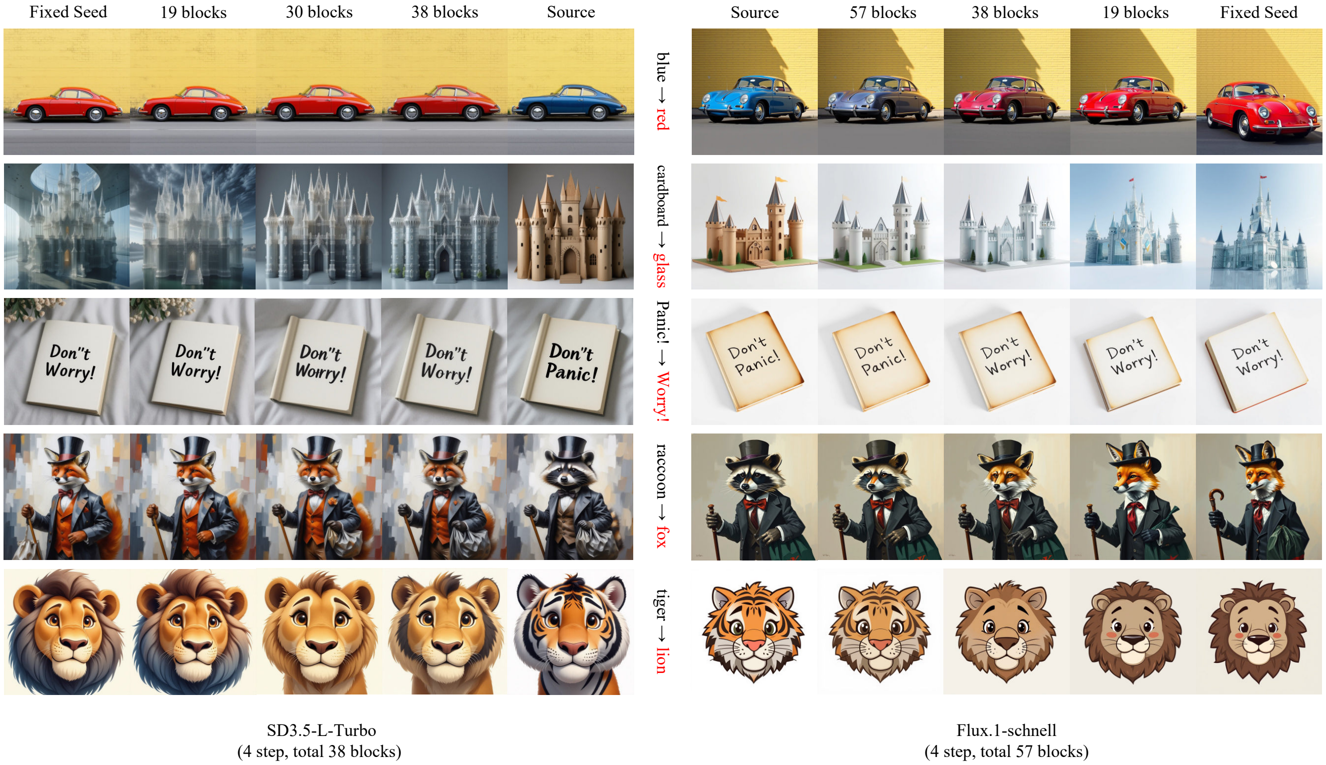


Figure A8. Qualitative results of SD3.5-L-Turbo (left) and Flux.1-schnell (right) demonstrating the impact of replacing block count (column) on edit strength, showing progression from source image through different block counts to fixed seed (corresponding to block count 0). Decreasing the number of replaced blocks strengthens the edit effect while reducing structural similarity and style to the source image. Local blending was not used to better focus on the impact of block replacement.

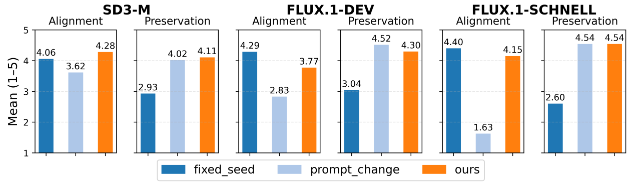


Figure A9. User study results comparing editing quality across different methods. Participants evaluated images based on target prompt alignment and source content preservation. Our method demonstrates superior balance between achieving desired edits while maintaining original image characteristics, outperforming both fixed seed generation (high prompt alignment, poor identity preservation) and prompt switching (good identity preservation, weak editing effects).

C.2. Impact of block selections on edit strength

As discussed in Sec. 4.3, the number of replacing blocks can serve as a hyperparameter to control edit strength. Fig. A8 presents ablation studies on two MM-DiT few-step models, where we varied the number of replacing blocks from the initial block. A notable observation is that some generated images exhibit excessive similarity to the source image, even with one timestep injection of our method, which constrains the model's editing capabilities. Given that further reduction in timesteps is impossible, we investigated adjusting the timestep scheduler to mitigate this similarity; however, this approach also proved ineffective in addressing the limitations. In this context, block control emerges as a particularly effective solution for 4-step distilled models, SD3.5-L-Turbo and Flux.1-schnell, with the latter showing a more pronounced effect. Through empirical investigation, we find that replacing blocks 38 and 30 yields favorable results for Flux.1-schnell and SD3.5-L-Turbo, respectively.

D. User Study and Additional Qualitative Results

To address the limitations of LPIPS and CLIP scores in capturing nuanced edit quality, we conducted a comprehensive user study using samples from Tab. 2. The study evaluated three widely-used models (SD3-M, Flux.1-dev, and Flux.1-schnell) with at least 30 participants per model (96 participants total). For fair comparison, all results were generated using purely $\mathbf{q}_i$, $\mathbf{k}_i$ replacement without manual per-sample local blending, though it would further enhance outcomes. Our findings reveal that our method uniquely balances strong target alignment (akin to direct generation, which sacrifices preservation contrastingly) with content preservation (comparable to prompt-change, which however fails to implement the edit).

Beyond the user study validation, we present extensive qualitative results (Fig. A11, Fig. A12) from our benchmark experiments discussed in Sec. 6, Tab. 2, and Fig. A9. These results showcase the robustness of our core approach across diverse editing scenarios using only input projection replacement, without hyperparameter tuning for individual cases. Additionally, Fig. A13 demonstrates exemplary cases where local blending was applied to achieve enhanced visual quality and editing precision.

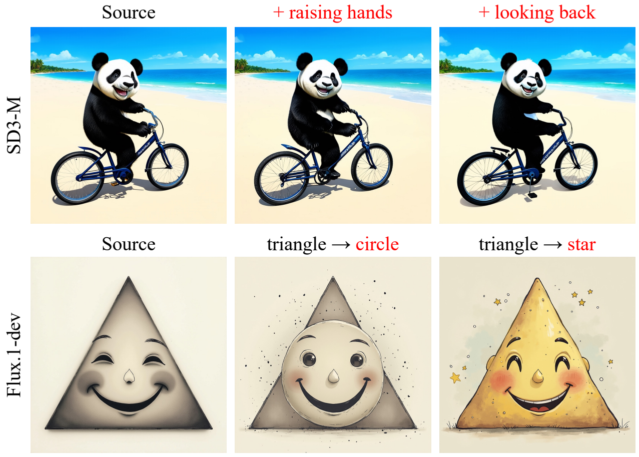


Figure A10. Limitations of our method. As noted in Sec. 7, our method is unable to achieve identity-preserving non-rigid transformations like those demonstrated in MasaCtrl [4]. Despite exploring various strategies to effectively modify self-attention related components, we found it challenging to develop a robust method that could replicate the capabilities of prior works using U-Net backbones. Since our method operates by replacing target input projections with source input projections during early timesteps, modifying low-level structures that are determined in these stages can occasionally be challenging, particularly in few cases involving colors and rough geometric layouts. While manually adjusting replacing timesteps can mitigate this issue, we leave the development of a more systematic solution for future work.

E. Comparison with Other Recent Works

To contextualize our contribution, this section discusses several recent and concurrent research efforts that have emerged alongside the effectiveness of modern MM-DiT architectures and RF formulations. These works often aim to improve real-image editing by building upon more accurate inversion methods and manipulating internal features. For instance, RF-Solver [63] uses a Taylor expansion to derive a more precise ODE solution and reduce inversion errors. Similarly, FireFlow [13] proposes an efficient few-step numerical solver that achieves second-order accuracy at first-order computational cost by reusing intermediate velocities. Both of these works also suggest swapping value features from the source branch into the target branch's self-attention layers to better preserve original content. Other approaches like FluxSpace [9] define a semantic representation space from the projected value features post-attention, and interpolate within this space to add controls. Additionally, StableFlow [2] identifies that not all layers in MM-DiT contribute equally to image formation and proposes a method to find a sparse set of "vital layers" crucial for the output. This concept, while not a direct equivalent, resonates with our finding of selecting optimal blocks with less noisy attention maps.

In contrast to these approaches, our paper's fundamental contribution is an architectural analysis of MM-DiT's attention mechanisms. We explore how principles from prior models like U-Net can be effectively transferred, leading to a precise, prompt-based editing method tailored to modern MM-DiT architectures. This attention-centric analysis provides unique insights orthogonal to the aforementioned methods, enabling detailed edits through attention control that function even without inversion, yet yielding superior results when paired with techniques like RF inversion as shown in our experiments.

F. Limitations and Future Directions

Our approach enables precise attention control through optimal transformer block selection and targeted input projection modifications. However, two limitations persist: the need for empirical parameter tuning in local blending and the inability to support identity-preserving non-rigid transformations (Fig. A10). Beyond these technical limitations, we observe promising potential in applying these models to visual grounding and segmentation tasks, given their ability to capture abstract attributes transcending conventional object boundaries. We leave these challenges as potential directions for future research.

G. Used Prompts

In this section, we provide the list of prompts used to generate the main paper figures, where they were not explicitly stated in the text. Relevant codes for reproducing our results will be open-sourced upon publication. Due to space constraints, additional prompts used for benchmarking and supplemental figures will be available in our public repository.

Figure 1.

- Source 1: *"beautiful oil painting of a steamboat in a river in the afternoon. On the side of the river is a large brick building with a sign on top that says 'SD3"'*
- Target 1: *"beautiful oil painting of a steamboat in a river in the afternoon. On the side of the river is a large brick building with a sign on top that says 'FLUX"'*
- Source 2: *"Detailed pen and ink drawing of a happy giraffe butcher selling meat in its shop"*
- Target 2: *"Detailed pen and ink drawing of a happy dragon butcher selling meat in its shop"*
- Source 3: *"A photograph of the inside of a subway train. There are frogs sitting on the seats. One of them is reading a newspaper. The window shows the river in the background"*
- Target 3: *"A photograph of the inside of a subway train. There are rabbits sitting on the seats. One of them is reading a newspaper. The window shows the river in the background"*
- Source 4: *"a guy in the forest with a sword and shield, fighting a dragon, holding a large sign 'help me"'*
- Target 4: *"a guy in the forest with a sword and shield, fighting a dragon, holding a large sign 'Please don't kill me"'*
- Source 5: *"A crab made of cheese on a plate"*
- Target 5: *"A cartoon-style drawing of a crab made of cheese on a plate"*
- Source 6: *"translucent pig, inside is a smaller pig"*
- Target 6: *"translucent whale, inside is a smaller whale"*
- Source 7: *"A 4K DSLR image of a Hound dog dressed in a finely tailored houndstooth check suit with bold, oversized patterns standing on a perfectly manicured grassy field holding a beautifully crafted banner that says 'Go Puppy Team!' "*
- Target 7: *"A 4K DSLR image of a Zebra dressed in a finely tailored zebra-striped suit with bold, oversized patterns standing on a perfectly manicured grassy field holding a beautifully crafted banner that says 'Go Zebra Team!' "*
- Source 8: *"A mischievous ferret with a playful grin squeezes itself into a large glass jar, surrounded by colorful candy. The jar sits on a wooden table in a cozy kitchen, and warm sunlight filters through a nearby window"*
- Target 8: *"A mischievous lion with a playful grin squeezes itself into a large glass jar, surrounded by colorful candy. The jar sits on a wooden table in a cozy kitchen, and warm sunlight filters through a nearby window"*

Figure 3.

- *"a panda riding a bicycle on the beach under blue sky"*

Figure 4.

- *"a photograph of a fiddle next to a basketball on a ping pong table"*

Figure 6.

- *"a cute tiger driving a sports car under starry night with blue moon in new york"*

Figure 8.

- Source: *"a drawing of a series of musical notes wrapped around the Earth"*
- Target: *"a drawing of a series of musical notes wrapped around the Moon"*

Figure 10.

- Source: *"a panda riding a bicycle on the beach under blue sky"*
- Target 1: *"a princess with a crown riding an elephant on*

the beach under blue sky"

- Target 2: *"a squirrel with a baseball cap riding a blue motorbike on the beach under blue sky"*
- Target 3: *"a cute hamburger with fried chicken legs riding a green motorbike in the grand canyon under blue sky"*

Figure 11.

- Source: *"A whimsical scene featuring a playful hybrid creature: a hippopotamus with golden, crispy waffle-textured skin, lounging in a surreal habitat blending water and breakfast elements like giant utensils and plates."*
- Target: *"A whimsical scene featuring a playful hybrid creature: an elephant with golden, crispy waffle-textured skin, lounging in a surreal habitat blending water and breakfast elements like giant utensils and plates."*

Figure 12.

- Source 1: *"beautiful oil painting of a steamboat in a river in the afternoon. On the side of the river is a large brick building with a sign on top that says ‘SD3”’*
- Target 1: *"beautiful oil painting of a steamboat in a river in the afternoon. On the side of the river is a large brick building with a sign on top that says ‘FLUX”’*
- Source 2: *"a cat sitting on a stairway railing"*
- Target 2: *"a squirrel sitting on a stairway railing"*

Figure 15.

- Source 1: *"a grandmother reading a book to her grandson and granddaughter"*
- Target 1: *"a grandmother reading a holographic storybook to her grandson and granddaughter in a floating space station"*
- Source 2: *"three green peppers"*
- Target 2: *"three red peppers"*
- Source 3: *"A close-up high-contrast photo of Sydney Opera House sitting next to Eiffel tower, under a blue night sky of roiling energy, exploding yellow stars, and radiating swirls of blue"*
- Target 3: *"A close-up high-contrast photo of Sydney Opera House sitting next to Eiffel tower, under a purple night sky of roiling energy, exploding yellow stars, and radiating swirls of purple"*
- Source 4: *"a comic about two cats doing research"*
- Target 4: *"a comic about two cats doing quantum physics research in a lab full of glowing experiments"*
- Source 5: *"a cartoon of a bear birthday party"*
- Target 5: *"a cartoon of a panda birthday party"*
- Source 6: *"a cat patting a crystal ball with the number 7 written on it in black marker"*
- Target 6: *"a cat patting a crystal ball with the number 13 written on it in black marker"*

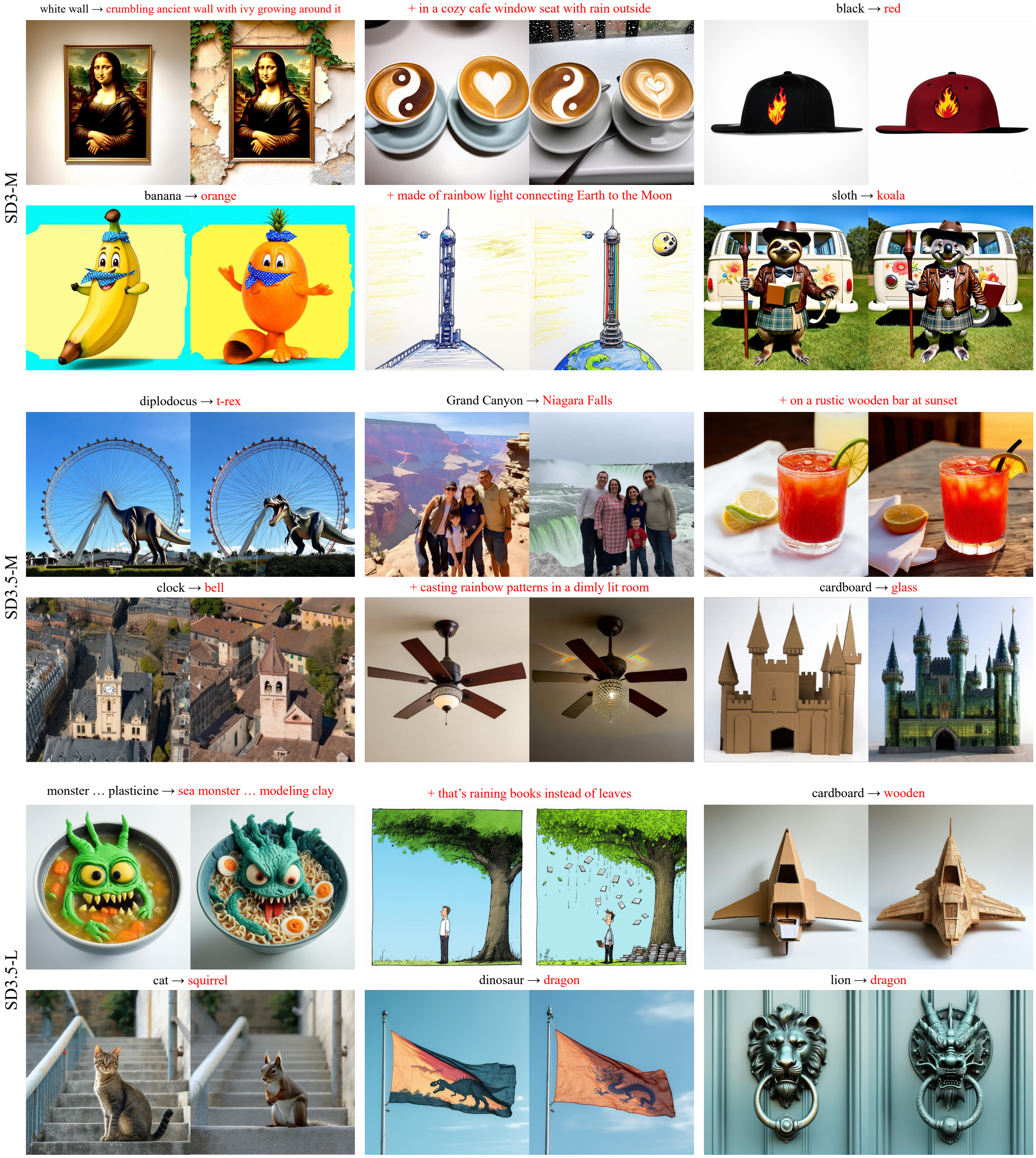

SD3-M
white wall → crumbling ancient wall with ivy growing around it
+ in a cozy cafe window seat with rain outside
black → red
banana → orange
+ made of rainbow light connecting Earth to the Moon
sloth → koala
SD3.5-M
diplodocus → t-rex
Grand Canyon → Niagara Falls
+ on a rustic wooden bar at sunset
clock → bell
+ casting rainbow patterns in a dimly lit room
cardboard → glass
SD3.5-L
monster … plasticine → sea monster … modeling clay
+ that's raining books instead of leaves
cardboard → wooden
cat → squirrel
dinosaur → dragon
lion → dragon

Figure A11. Additional qualitative results from experiments reported in Tab. 2, demonstrating edits across diverse model variants. All results shown use only input projection replacements ($\mathbf{q}_i$, $\mathbf{k}_i$), without local blending operation.

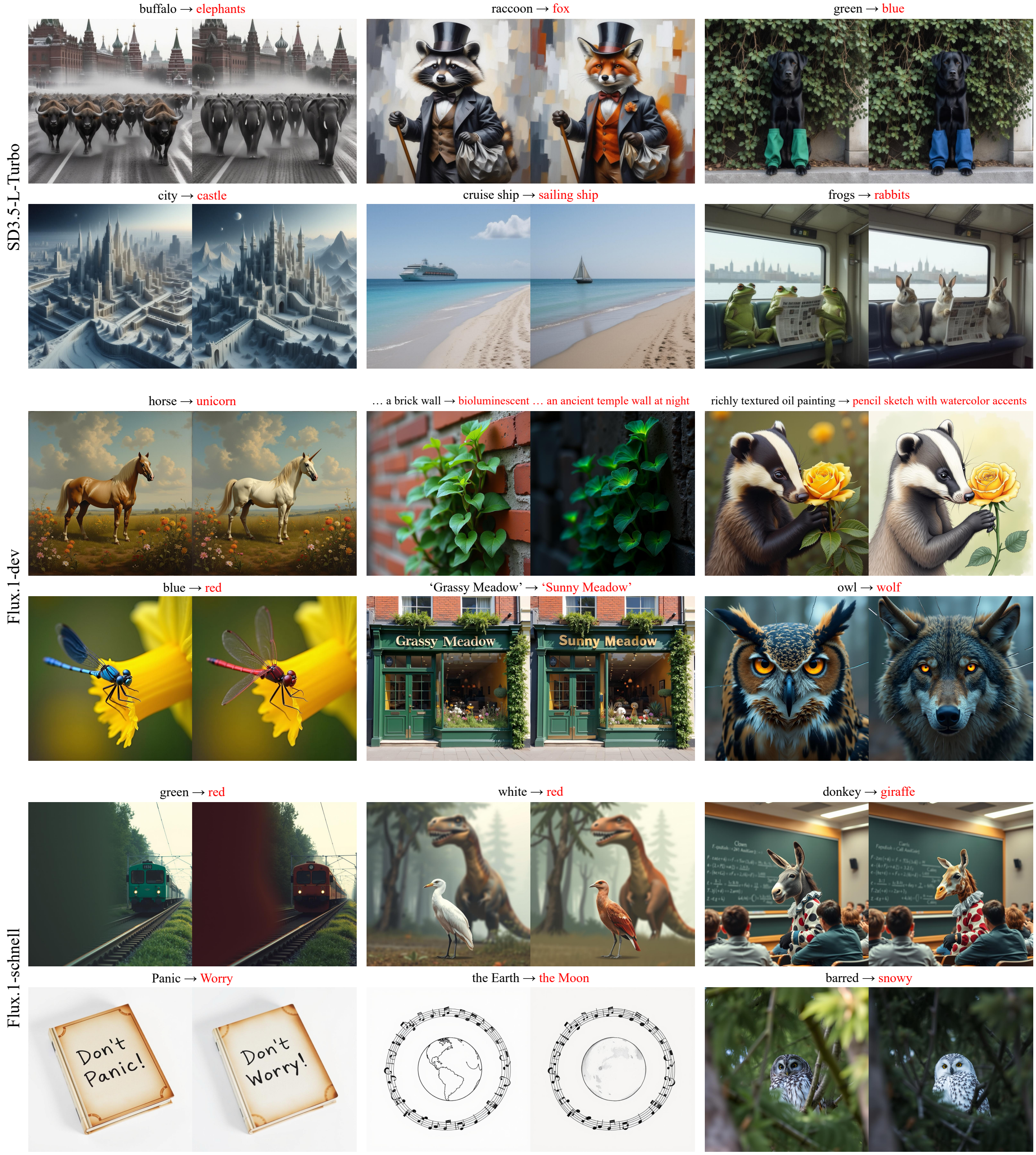


Figure A12. Additional qualitative results from experiments reported in Tab. 2, demonstrating edits across diverse model variants. All results shown use only input projection replacements ($\mathbf{q}_i$, $\mathbf{k}_i$), without local blending operation.

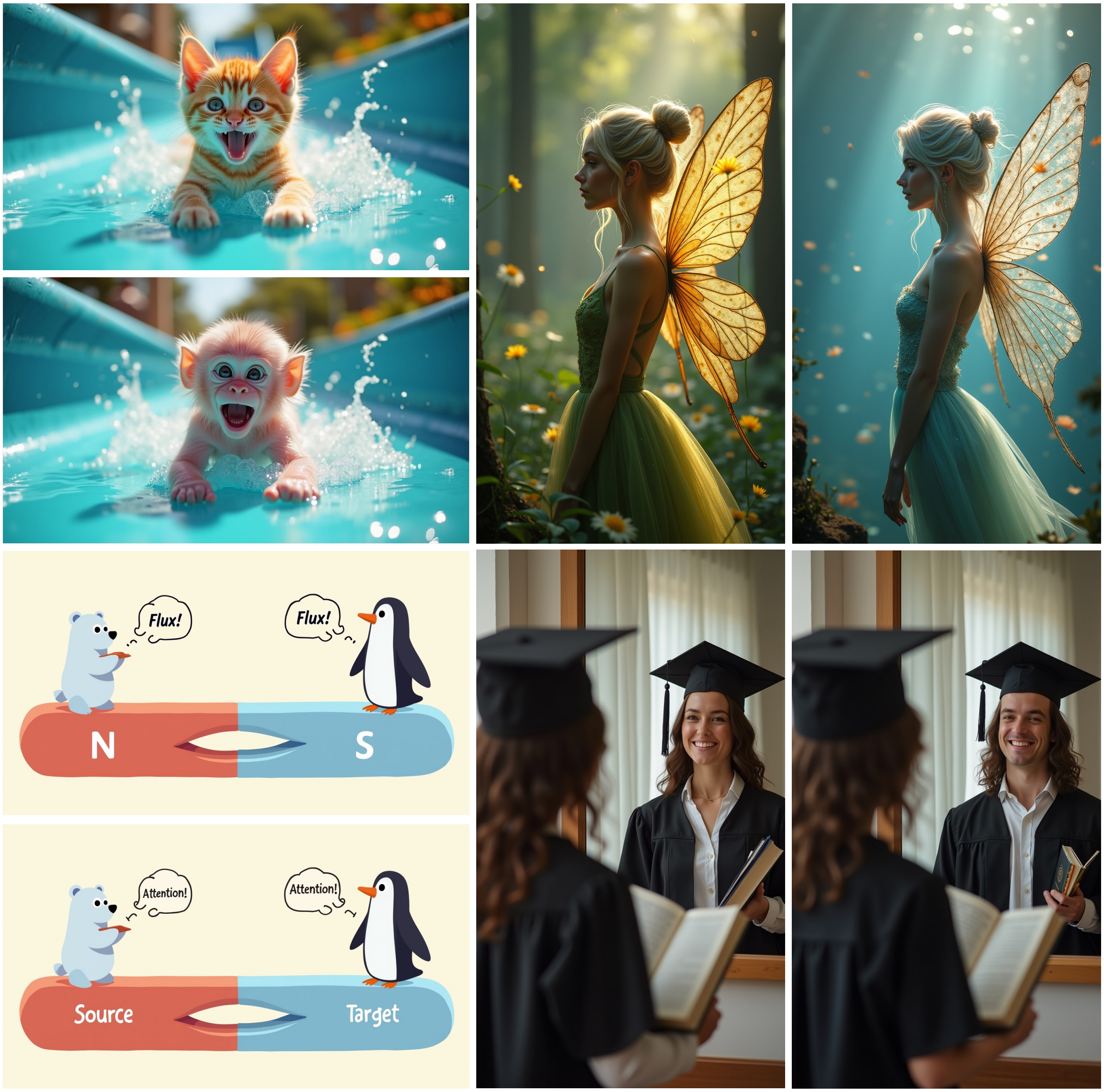


Figure A13. Additional qualitative results of our method, showcasing various editing scenarios: (a) changing 'orange kitten' into 'pink monkey', (b) converting 'a forest fairy' into 'an ocean fairy', (c) modifying text from 'N, S, and Flux!' to 'Source, Target, and Attention!', and (d) changing the identity of subjects, such as transforming a 'grown woman' into a 'grown man'.

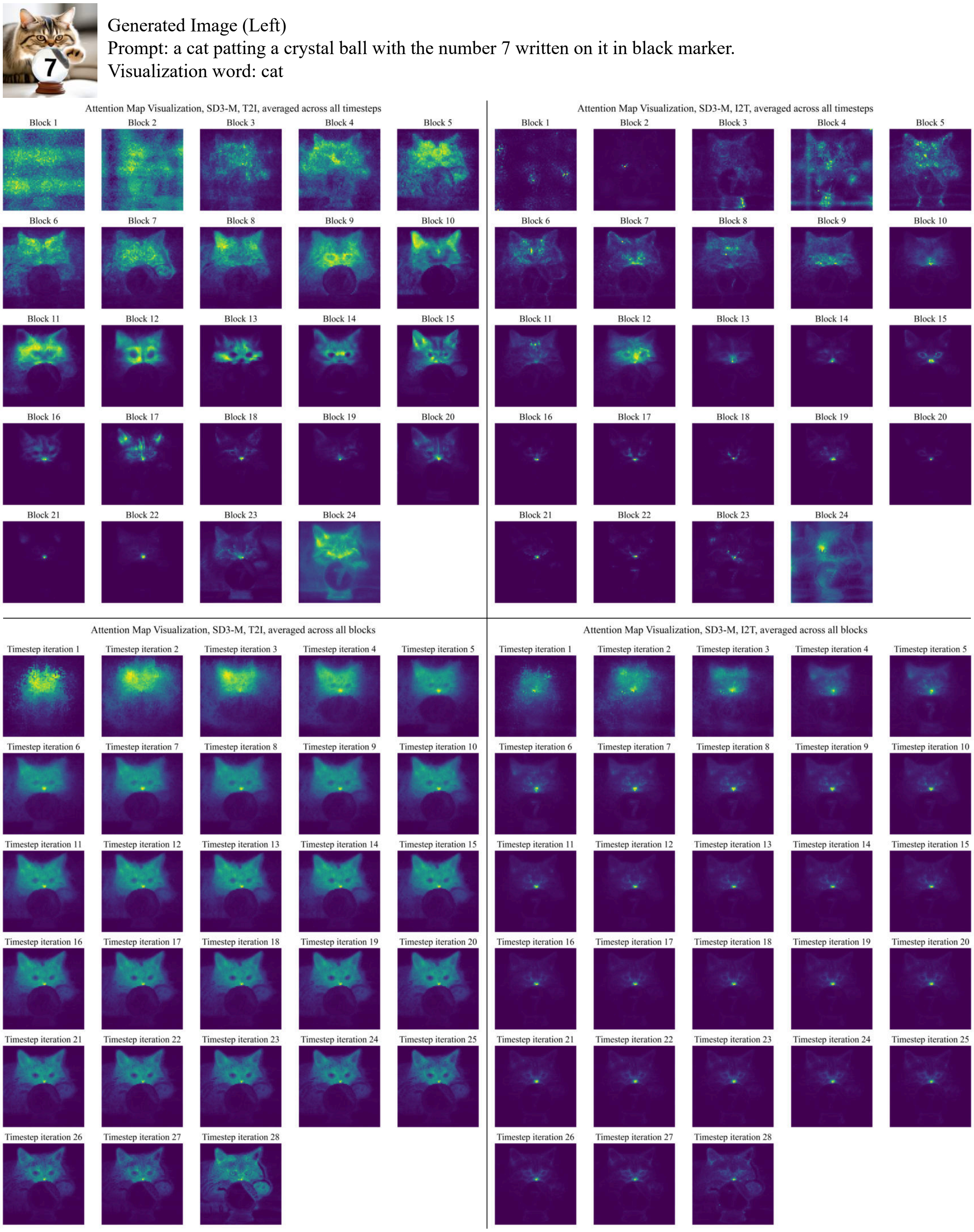


Figure A14. Visualization of T2I (left) and I2T (right) attention maps in SD3-M. Upper rows show per-block attention maps (averaged across 28 timesteps), while lower rows show per-timestep attention maps (averaged across all blocks). T2I portions generally capture semantic concepts more effectively, though certain blocks exhibit significant noise. Timestep-wise analysis reveals that image structure and layout are primarily established in early denoising steps, supporting our approach of attention map replacement during only the first 20% of timesteps to preserve original image characteristics. Best viewed zoomed in.

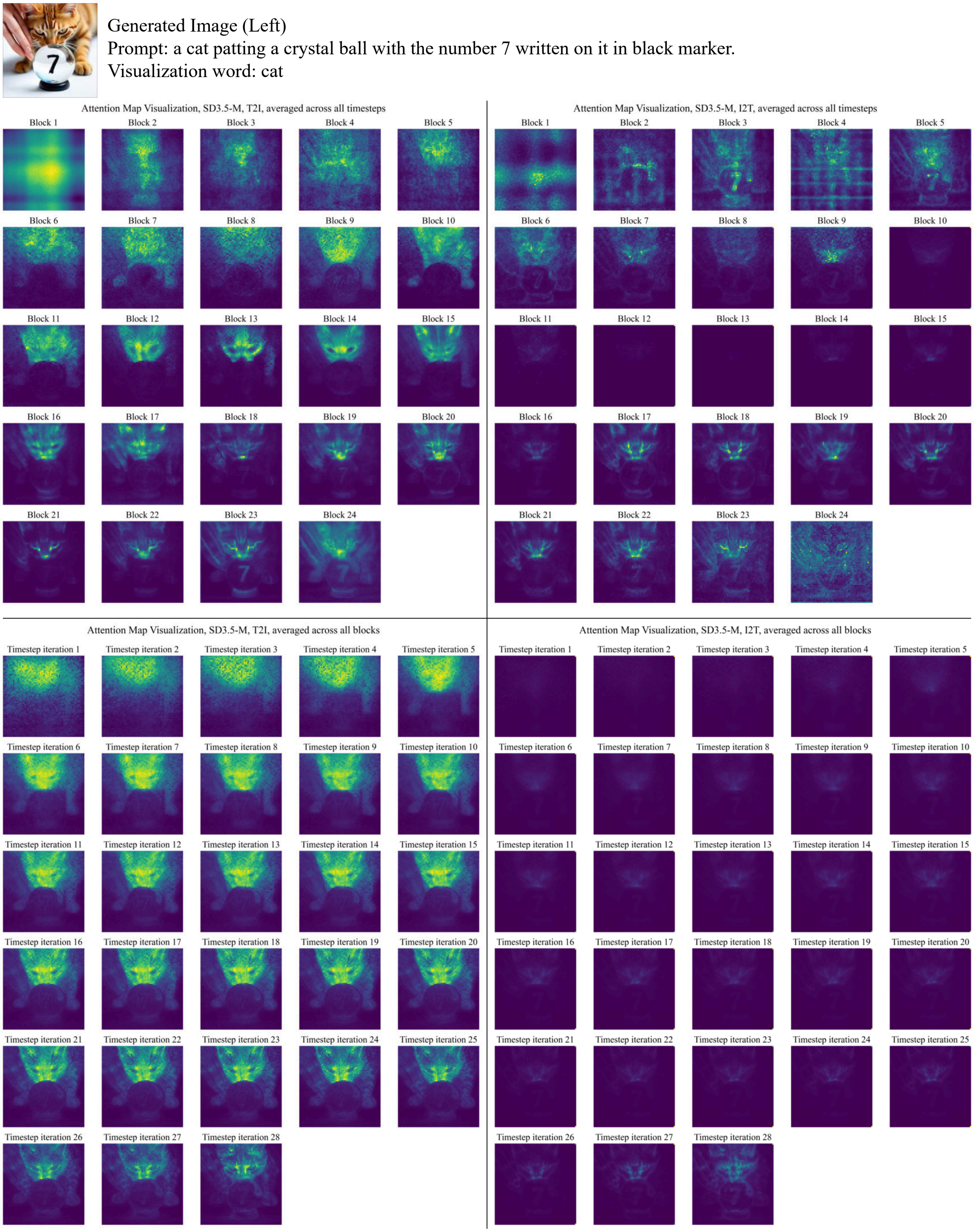


Figure A15. Visualization of T2I (left) and I2T (right) attention maps in SD3.5-M, averaged across timesteps (upper) and across transformer blocks (lower). The visualization format and observed patterns are mostly consistent with Fig. A14. Best viewed zoomed in.

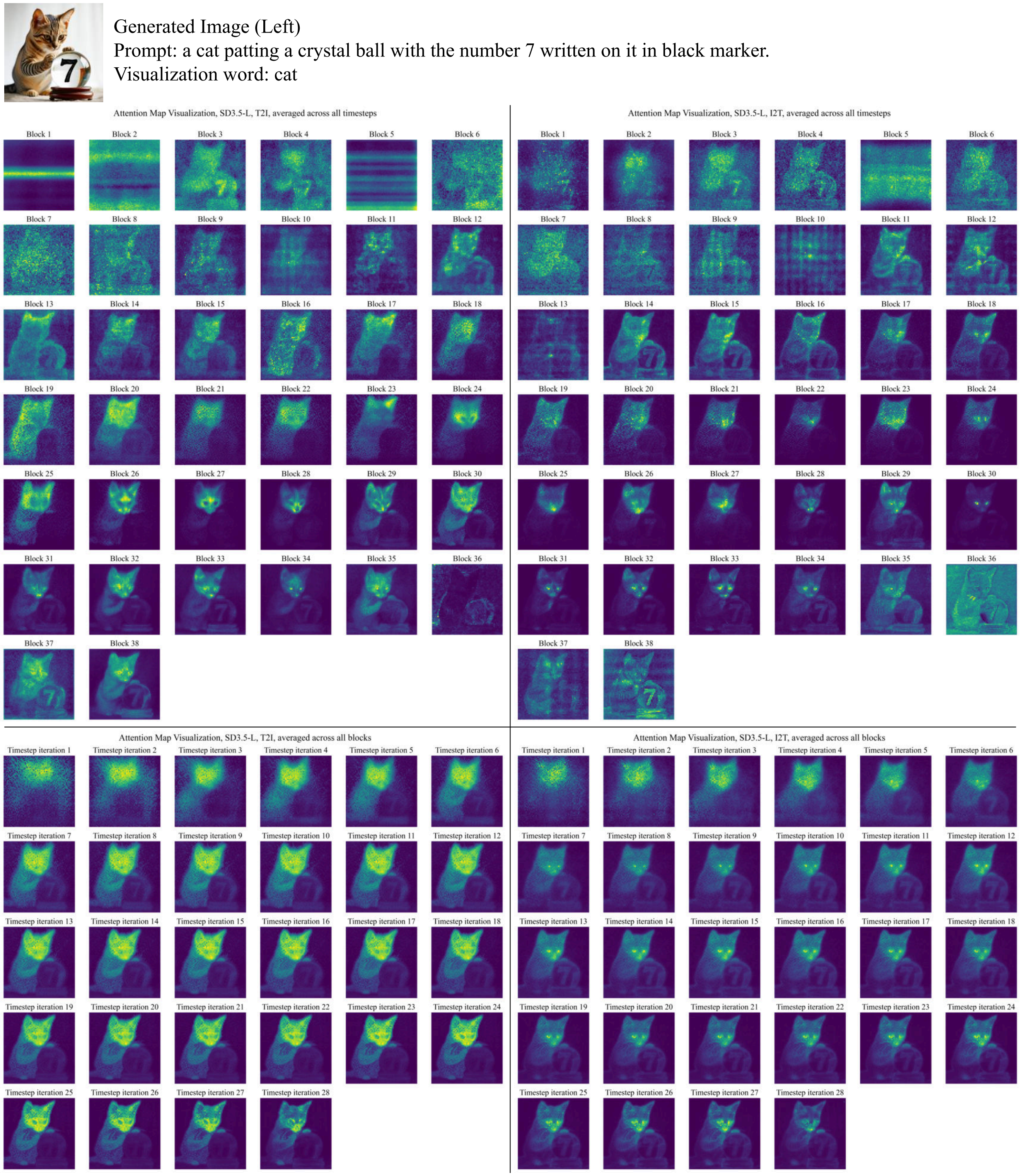


Figure A16. Visualization of T2I (left) and I2T (right) attention maps in SD3.5-L, averaged across timesteps (upper) and across transformer blocks (lower). The visualization format and observed patterns are mostly consistent with Fig. A14, except that we observe noisier attention maps. Best viewed zoomed in.

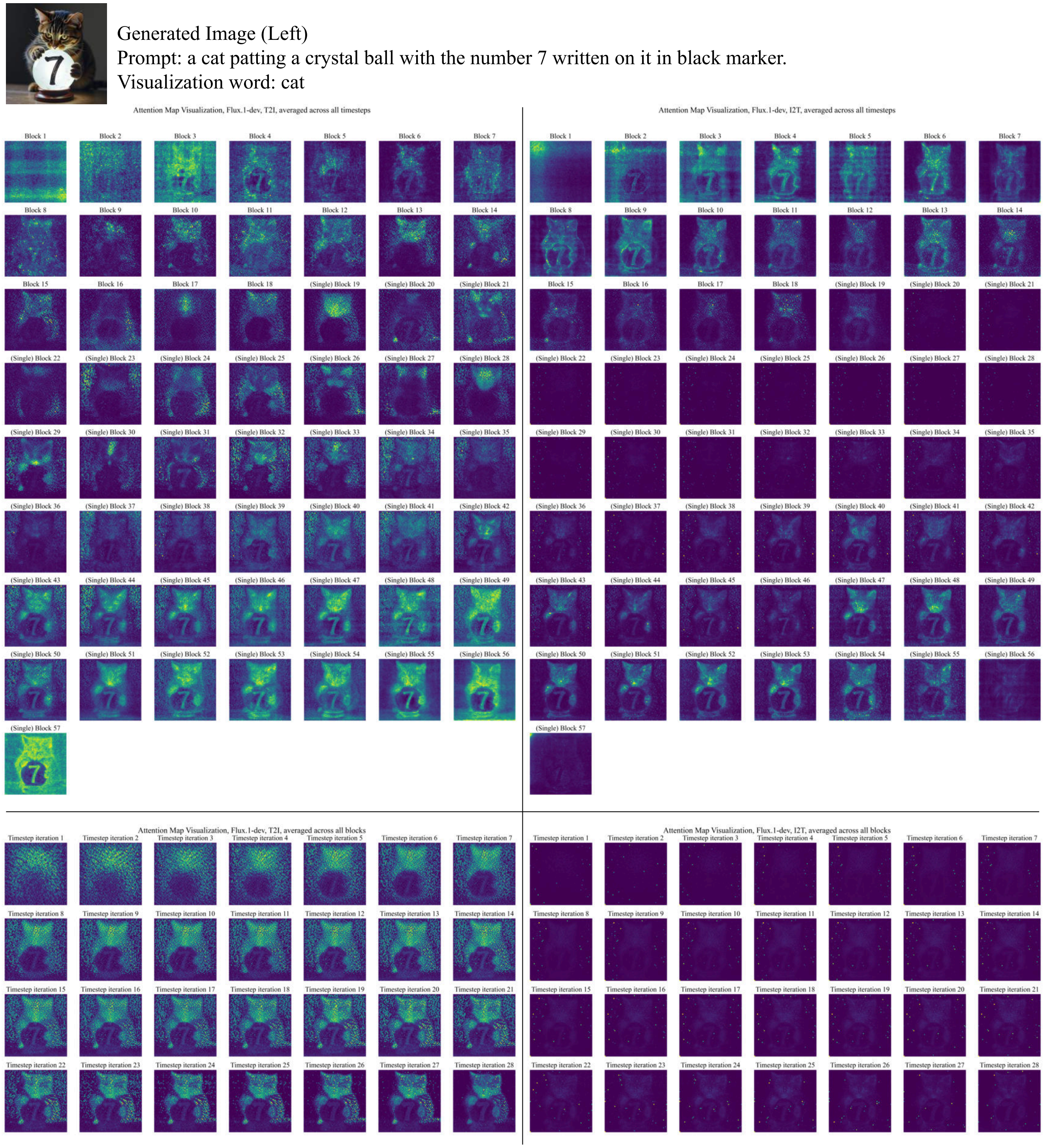
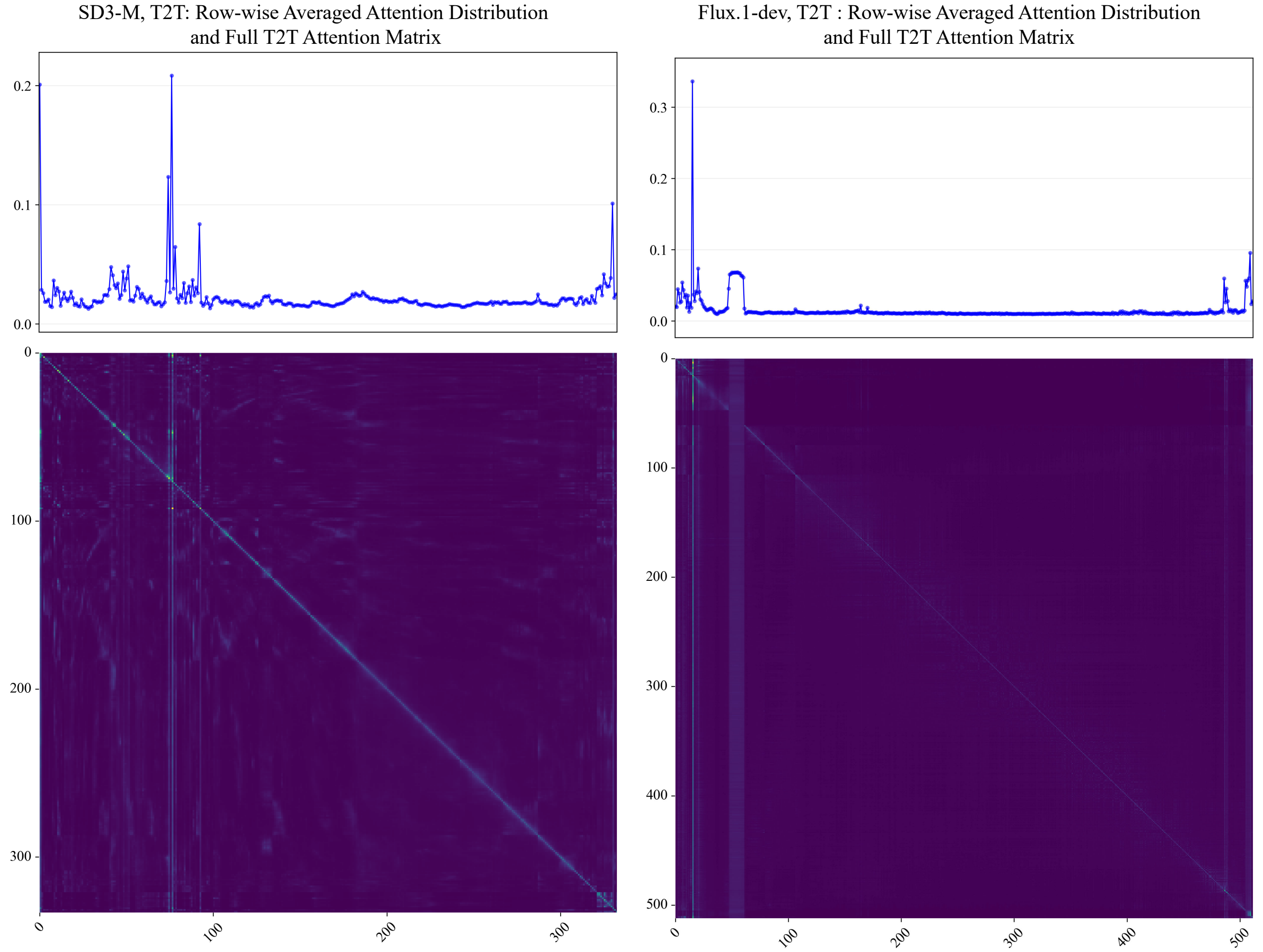


Figure A17. Visualization of T2I (left) and I2T (right) attention maps in Flux.1, averaged across timesteps (upper) and across transformer blocks (lower). The visualization format and observed patterns are mostly consistent with Fig. A14. It is worth noting that even in single-branch transformers, geometric and spatial patterns are preserved, indicating the preservation of information for each domain. Attention maps appear noisy in some blocks. Best viewed zoomed in.



SD3-M, T2T: Row-wise Averaged Attention Distribution and Full T2T Attention Matrix

Flux.1-dev, T2T : Row-wise Averaged Attention Distribution and Full T2T Attention Matrix

Figure A18. Visualization of the T2T portion of the attention maps in SD3-M (left) and Flux.1-dev (right). The heatmaps mostly show diagonal patterns, with stronger signals from special tokens. Above each heatmap, we present row-wise averaged attention values as line plots to better highlight the relative values among column indices.