Stable Flow: Vital Layers for Training-Free Image Editing Supplementary Material

Omri Avrahami[1,2] Or Patashnik[1,3] Ohad Fried[4] Egor Nemchinov[1]
Kfir Aberman[1] Dani Lischinski[2] Daniel Cohen-Or[1,3]
[1]Snap Research [2]The Hebrew University of Jerusalem [3]Tel Aviv University [4]Reichman University

1. Implementation Details

In Section 1.1, we start by providing implementation details for our method. Next, in Section 1.2, we provide the implementation details for the baselines we compared our method against. Later, in Section 1.3, we provide the implementation details for the automatic evaluations dataset and metrics. Finally, in Section 1.4 we provide the full details of the user study we conducted.

1.1. Method Implementation Details

As described in Section 3.1 of the main paper, we started by collecting a dataset of $k = 64$ text prompts using ChatGPT [15]. We instructed it to generate text prompts describing a diverse set of objects in different environments, with the focus on one main object. Then, we sampled k seeds denoted by S and used them to generate k corresponding images G_{ref}. Next, for each layer l, we bypass it by taking only the residual connection values. For each bypass, we generate k images using the same seed set S demoted by G_l. All the images were generated using Euler sampler in 15 steps and a guidance scale of 3.5.

Next, to evaluate the effect of each layer l on the final result, we compared the generated images G_l with their corresponding images G_{ref} using the DINOv2 [16] perceptual similarity metric. We term the layers that effect the generated image the most (*i.e.*, the layers with the lowest perceptual similarity) as vital layers, while the rest of the layers as non-vital layers. We found that the vital layers in the FLUX.1-dev model [11] are $[0, 1, 2, 17, 18, 25, 28, 53, 54, 56]$. For visualization results, please refer to Section 2.6. We empirically found that layer 2 can be removed from this set. In addition, the vital layers for the Stable Diffusion 3 (SD3) [7] model vital layers are: $[0, 7, 8, 9]$. For more details, please refer to Section 2.7.

In addition, as mentioned in Section 3.2 of the main paper, We adapt the self-attention injection mechanism, previously to be effective for image and video editing [5, 21] in UNet-based diffusion models, to the DiT-based FLUX architecture. Since each DiT layer processes a sequence of image and text embeddings, we propose generating both the reference image x and generated image $\hat{x}$ in parallel while *selectively replacing* the image embeddings of $\hat{x}$ with those of x, but only within the vital layers set. A full visualization can be found in Figure 1.

Lastly, the variance list of the perceptual similarity of the different layers, as explained in Section 3.1 of the main paper, is as follows: [0.222, 0.041, 0.076, 0.08, 0.123, 0.101, 0.135, 0.124, 0.112, 0.105, 0.097, 0.12, 0.118, 0.086, 0.116, 0.067, 0.065, 0.116, 0.146, 0.065, 0.098, 0.061, 0.076, 0.077, 0.072, 0.086, 0.069, 0.067, 0.081, 0.091, 0.074, 0.062, 0.061, 0.044, 0.04, 0.054, 0.036, 0.038, 0.037, 0.04, 0.066, 0.04, 0.034, 0.044, 0.044, 0.031, 0.033, 0.036, 0.03, 0.032, 0.026, 0.026, 0.026, 0.079, 0.039, 0.037, 0.026].

1.2. Baselines Implementation Details

As explained in Section 4.1 of the main paper, we compare our method against the following baselines: SDEdit [13], P2P+NTI [9, 14], Instruct-P2P [4], MagicBrush [23], and MasaCTRL [5]. We reimplement SDEdit using the FLUX.1-dev model [11], and use the official implementation for the rest of the baselines.

We adapt the text prompts based on the baseline type: for SDEdit [13], P2P+NTI [9, 14], and MasaCTRL [5], we used the standard text prompt describing the desired edited scene (*e.g.*, *"A photo of a man with a red hat"*). For the instruction-based baselines Instruct-P2P [4] and MagicBrush [23] we adapted the style to fit an instructional format (*e.g.*, *"Make the person wear a red hat"*).

We used the following third-party implementations in this project:

- **FLUX.1-dev** model [11] HuggingFace Diffusers [19] implementation at `https://github.com/huggingface/diffusers`
- **P2P+NTI** [9, 14] official implementation at `https://github.com/google/prompt-to-prompt`
- **Instruct-P2P** [4] official implementation at `https://github.com/timothybrooks/instruct-pix2pix`

Figure 1. **Attention Injection.** We adapt the self-attention injection mechanism, previously shown effective for image and video editing in UNet-based diffusion models, to the DiT-based FLUX architecture. Since each DiT layer processes a sequence of image and text embeddings, we propose generating both the reference image x and generated image $\hat{x}$ in parallel while *selectively replacing* the attention *keys and values* that correspond to the image embeddings of $\hat{x}$ with those of x. This replacement is performed only within the vital layers set.

- **MagicBrush** [23] official implementation at https://github.com/OSU-NLP-Group/MagicBrush
- **MasaCTRL** [5] official implementation at https://github.com/TencentARC/MasaCtrl
- **DINOv2** [16] ViT-g/14 implementation by HuggingFace Transformers [20] at https://github.com/huggingface/transformers.
- **DINOv1** [6] ViT-B/16 implementation by HuggingFace Transformers [20] at https://github.com/huggingface/transformers.
- **CLIP** [18] ViT-L/14 implementation by HuggingFace Transformers [20] implementation at https://github.com/huggingface/transformers
- **LPIPS** [24] official implementation at https://github.com/richzhang/PerceptualSimilarity.

1.3. Automatic Metrics Implementation Details

As explained in Section 4.1 of the main paper, we prepare an evaluation dataset based on the COCO [12] validation dataset. We begin by filtering the dataset automatically to include at least one prominent non-rigid body. More specifically, we filter only images containing humans or animals that at least one of them is prominent enough, but not too small, *i.e.*, the prominent non-rigid body occupies at least 5% of the image but no more than 33%. Next, for each image, we apply various image editing tasks (non-rigid editing, object addition, object replacement, and scene editing) that take into account the prominent object from a list of different combinations, resulting in a total dataset of 3,200 samples. Examples of images from this dataset can be seen in Figure 4.

Table 1. **User Study Statistical Significance.** A binomial statistical test of the user study results suggests that our results are statistically significant (p-value $< 5\%$).

Ours vs	Prompt Adher. p-value	Image Pres. p-value	Realism p-value	Overall p-value
SDEdit [13]	$< 1e{-8}$	$< 1e{-8}$	$< 1e{-6}$	$< 1e{-8}$
P2P+NTI [9, 14]	$< 1e{-8}$	$< 1e{-8}$	$< 1e{-8}$	$< 6e{-8}$
Instruct-P2P [4]	$< 1e{-8}$	$< 1e{-8}$	$< 1e{-8}$	$< 2e{-4}$
MagicBrush [23]	$< 5e{-5}$	$< 1e{-8}$	$< 1e{-8}$	$< 1e{-8}$
MasaCTRL [5]	$< 1e{-8}$	$< 1e{-8}$	$< 1e{-8}$	$< 1e{-8}$

We evaluate the editing results using three metrics: (1) CLIP_{img} which measures the similarity between the input image and the edited image by calculating the normalized cosine similarity of their CLIP image embeddings. (2) CLIP_{txt} which measures the similarity between the edited image and the target editing prompt by calculating the normalized cosine similarity between the CLIP image embedding and the target text CLIP embedding. (3) CLIP_{dir} [8, 17] which measures the similarity between the direction of the prompt change and the direction of the image change.

1.4. User Study Details

As described in Section 4.2 of the main paper, we conducted an extensive user study using the Amazon Mechanical Turk (AMT) [2] platform, using automatically generated test examples, as explained in Section 1.3. We compared all the baselines with our method using a standard two-alternative forced-choice format. The users were given full instructions, as can be seen in Figure 2. Then, for each study trial, as shown in Figure 3, users were presented with an image

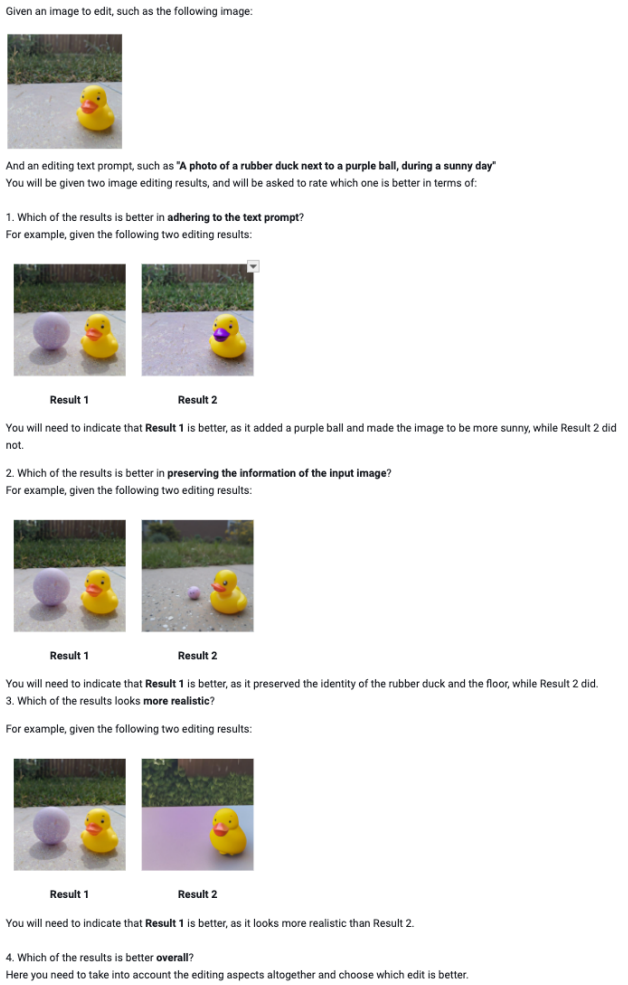


Figure 2. **User Study Instructions.** We provide the complete instructions for the user study we conducted using Amazon Mechanical Turk (AMT) [2] to compare our method with each baseline.

and an instruction *"Given the following input image of a {CATEGORY}"* where {CATEGORY} is the COCO category of the prominent object. The users were given two editing results — one from our method and one from the baseline, and were asked the following questions:

1. *"Which of the results is better in adhering to the text prompt {PROMPT}?"*, where {PROMPT} is the editing target prompt.
2. *"Which of the results is better in preserving the information of the input image?"*
3. *"Which of the results looks more realistic?"*
4. *"Which of the results is better in overall?"*

We collected five ratings per sample, resulting in 320 ratings per baseline, for a total of 1,920 responses. The time allotted per task was one hour, to allow raters to properly evaluate the results without time pressure. A binomial statistical test of the user study results, as presented in Table 1, suggests that our results are statistically significant (p-value $< 5\%$).

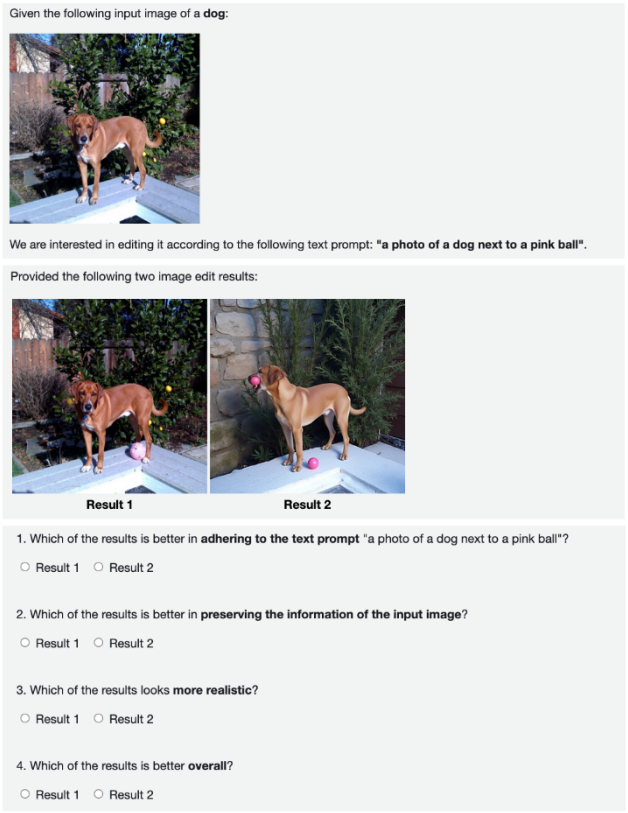


Figure 3. **User Study Trial.** We provide an example of a trial task in the user study conducted using Amazon Mechanical Turk (AMT) [2]. Users were asked four questions of a two-alternative forced-choice format. Complete instructions are shown in Figure 2.

2. Additional Experiments

In Section 2.1, we start by providing additional comparisons and results of our method. Then, in Section 2.2, we present experiments on using different perceptual metrics. Following that, in Section 2.4, we test the effect of different sizes for vital layer set. Next, in Section 2.5, we provide latent nudging experiments. Furthermore, in Section 2.6 we present a full visualization of our layer bypassing method. Finally, in Section 2.7, we test our method on the Stable Diffusion 3 backbone.

2.1. Additional Comparisons and Results

In Figure 4 we provide an additional qualitative comparison of our method against the baselines on real images extracted from the COCO [12] dataset, as explained in Section 4.1 in the main paper. As can be seen, SDEdit [13] struggles with preserving the object identities and backgrounds (*e.g.*, the

Figure 4. **Baselines Qualitative Comparison on Automatic Dataset.** As explained in Section 4.1 of the main paper, we compare our method against the baselines on real images extracted from the COCO [12] dataset. We find that SDEdit [13] struggles with preserving the object identities and backgrounds (*e.g.*, bear and chicken examples). P2P+NTI [9, 14] struggles with preserving object identities (*e.g.*, bear and person examples) and with adding new objects (*e.g.*, missing hat in the sheep example and missing ball in the elephant example). Instruct-P2P [4] and MagicBrush [23] struggle with non-rigid editing (*e.g.*, person raising hand). MasaCTRL [5] struggles with preserving object identities (*e.g.*, bear and person examples) and adding new objects (*e.g.*, sheep and cat examples). Our method, on the other hand, is able to adhere to the editing prompt while preserving the identities.

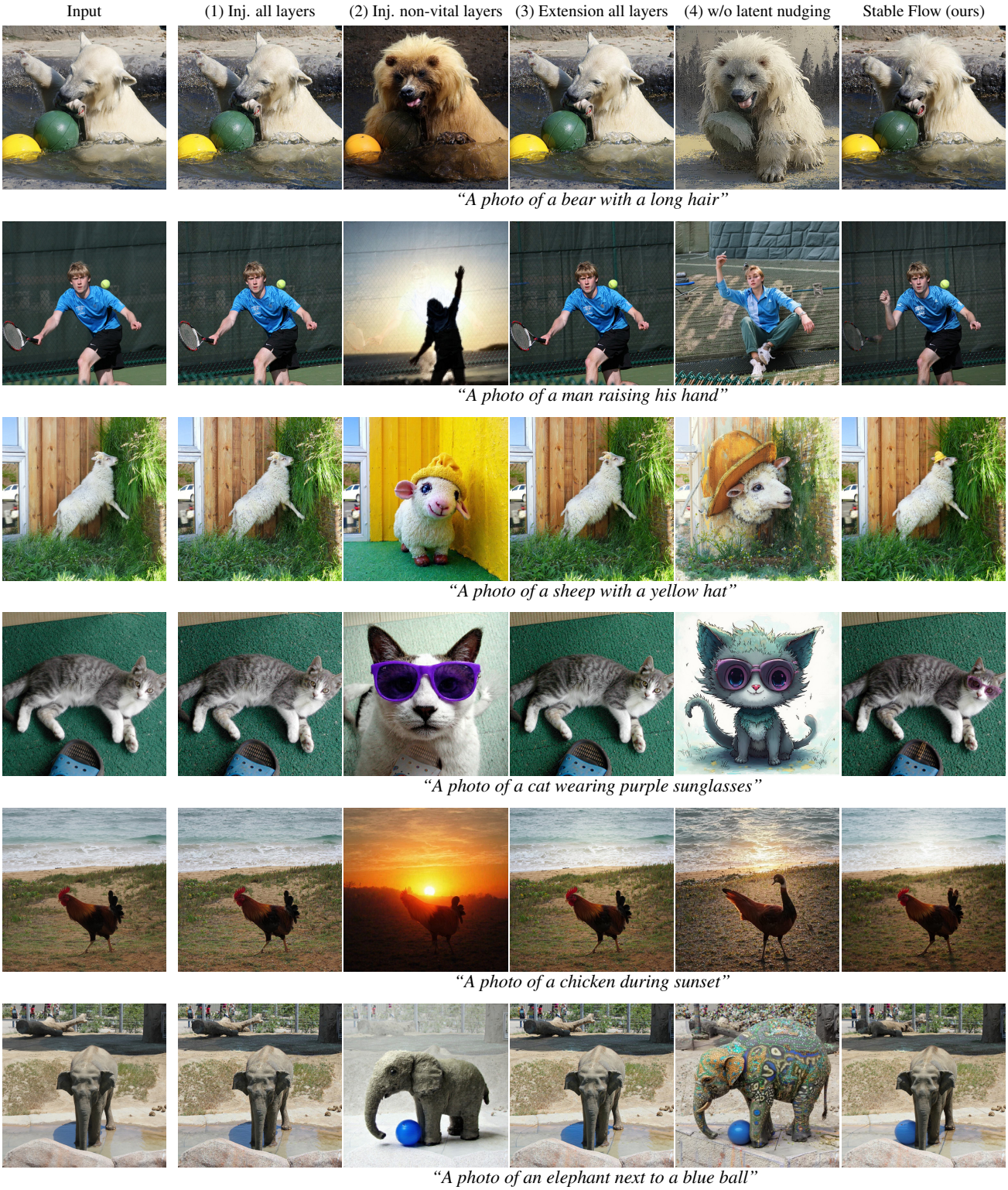
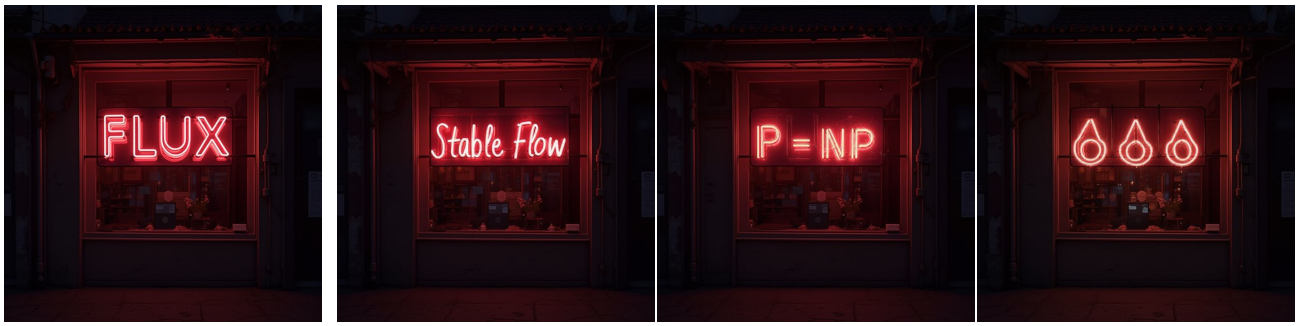


Figure 5. **Ablations Qualitative Comparison on Automatic Dataset.** As explained in Section 4.3 of the main paper, we compare our method against several ablation cases on real images extracted from the COCO [12] dataset. As can be seen, we found that (1) performing attention injection in all the layers or performing (3) an attention extension in all the layers encourages the model to directly copy the input image while neglecting the target prompt. In addition, (2) performing an attention extension in the non-vital layers or (4) removing the latent nudging reduces the input image similarity significantly.



Input | *"A 'Stable Flow' neon sign"* | *"A 'P = NP' neon sign"* | *"A neon sign of avocados"*

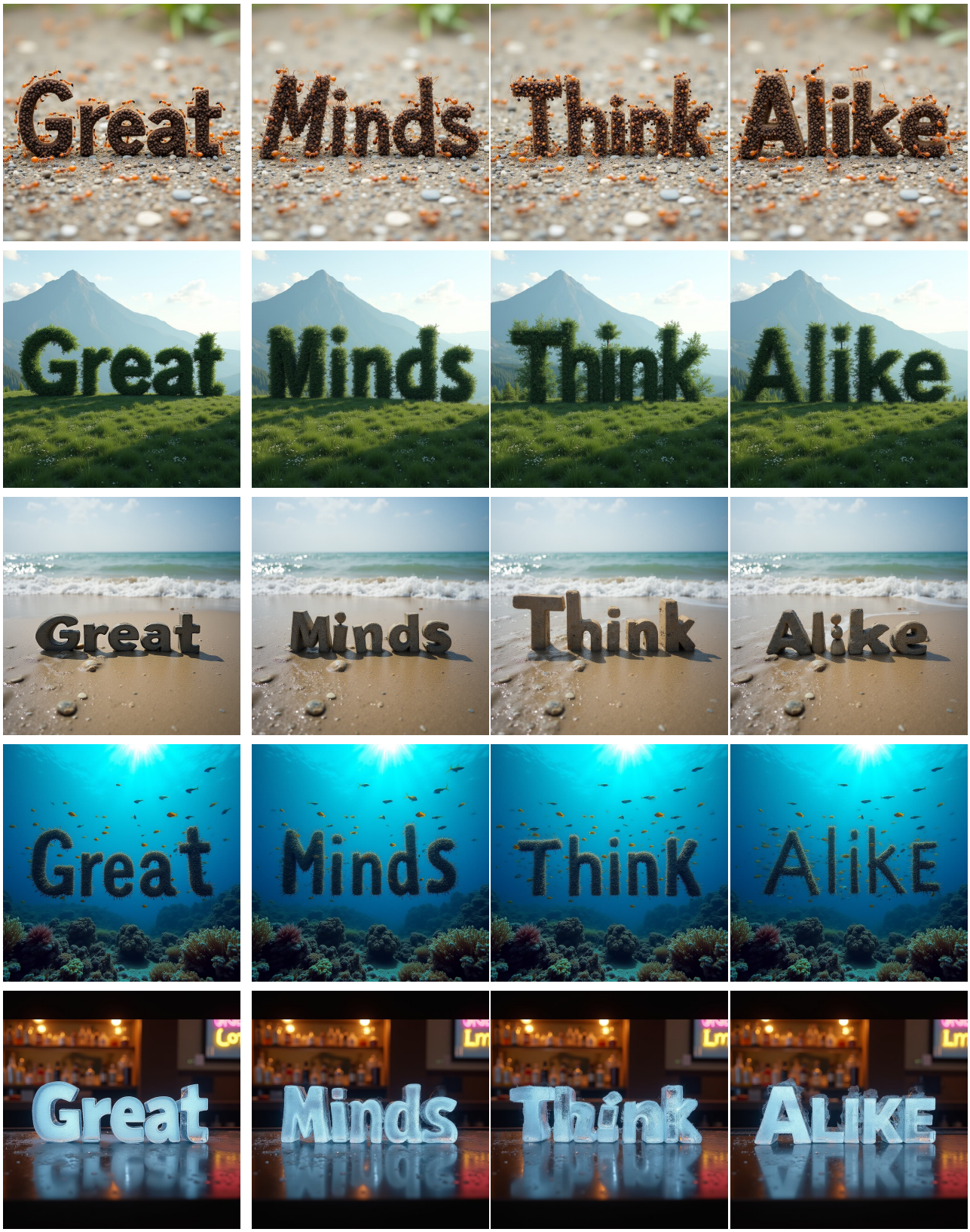
Input | *"A wooden lion"* | *"A wooden toilet"* | *"A wooden noodles bowl"*

Input | *"A hedgehog"* | *"A shark"* | *"A bird"*

Input | *"Jumping"* | *"Sitting"* | *"Putting its paw on a stone"*

Figure 6. **Additional Results.** We provide various editing results of our method. These different edits are done using the *same* vital layer set.

Figure 7. **Additional Results.** We provide various editing results of our method. These different edits are done using the *same* vital layer set.



Input *"Minds"* *"Think"* *"Alike"*

Figure 8. **Additional Results.** Given an input image that contain a text, our method cat edit the text while keeping the background and style.

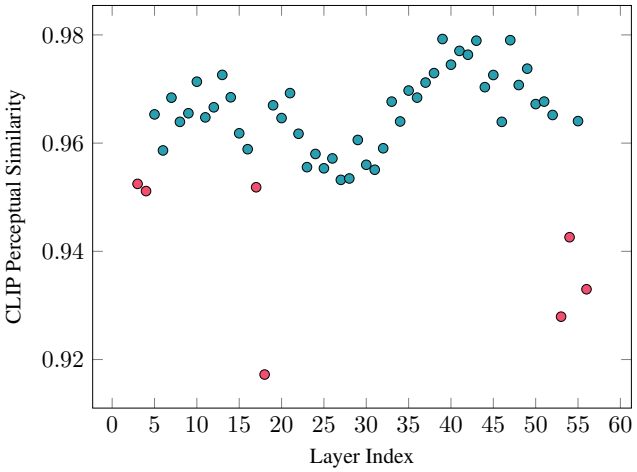


Figure 9. **Layer Removal Quantitative Comparison Using CLIP.** As explained in Section 2.2, we measured the effect of removing each layer of the model by calculating the *CLIP* [18] perceptual similarity between the generated images with and without this layer. Lower perceptual similarity indicates significant changes in the generated images. As can be seen, removing certain layers significantly affects the generated images, while others have minimal impact. Importantly, influential layers are distributed across the transformer rather than concentrated in specific regions. Note that the first vital layers were omitted for clarity (as their perceptual similarity approached zero).

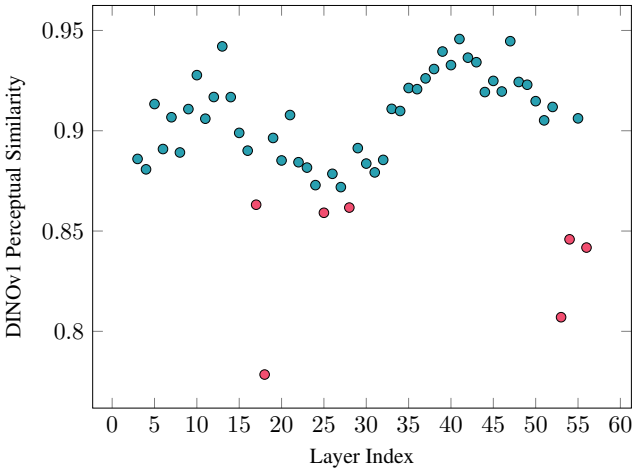


Figure 10. **Layer Removal Quantitative Comparison Using DINOv1.** As explained in Section 2.2, we measured the effect of removing each layer of the model by calculating the *DINOv1* [6] perceptual similarity between the generated images with and without this layer. Lower perceptual similarity indicates significant changes in the generated images. As can be seen, removing certain layers significantly affects the generated images, while others have minimal impact. Importantly, influential layers are distributed across the transformer rather than concentrated in specific regions. Note that the first vital layers were omitted for clarity (as their perceptual similarity approached zero).

bear and chicken examples). P2P+NTI [9, 14] struggles with preserving object identities (*e.g.*, the bear and person examples) and with adding new objects (*e.g.*, the missing hat in the sheep example and missing ball in the elephant example). Instruct-P2P [4] and MagicBrush [23] struggle with non-rigid editing (*e.g.*, the person raising hand example). MasaCTRL [5] struggles with preserving object identities (*e.g.*, the bear and person examples) and adding new objects (*e.g.*, the sheep and cat examples). Our method, on the other hand, is able to adhere to the editing prompt while preserving the identities.

Next, in Figure 5, we provide a qualitative comparison of the ablated cases that are explained in, Section 4.3 in the main paper. As can be seen, we found that (1) performing attention injection in all the layers or performing (3) an attention extension in all the layers, encourages the model to directly copy the input image while neglecting the target prompt. In addition, (2) performing an attention extension in the non-vital layers or (4) removing the latent nudging reduces the input image similarity significantly.

Finally, in Figures 6 and 7, we present additional image editing results using our method.

2.2. Different Perceptual Metrics

As explained in Section 3.1 of the main paper, we assess the impact of each layer by measuring the perceptual similarity between G_{ref} and G_ℓ using DINOv2 [16]. It raises the question of the importance of the specific perceptual [10] similarity metric when determining the vital layers.

To this end, we also experiment with different perceptual metrics: DINOv1 [6], CLIP [18], and LPIPS [24]. In Figures 9, 10 and 11 we plot the perceptual similarity per layer for each of these metrics. The vital layers, ordered by vitality, as defined in Equation 1 of the main paper, for each metric are:

- DINOv2 — $[1, 0, 2, 18, 53, 28, 54, 17, 56, 25]$.
- DINOv1 — $[1, 0, 2, 18, 53, 56, 54, 25, 28, 17]$.
- CLIP — $[2, 0, 1, 18, 53, 56, 54, 4, 17, 3]$.
- LPIPS — $[0, 1, 2, 18, 17, 56, 53, 54, 6, 4]$.

As can be seen, the vital set V is equivalent for DINOv2 and DINOv1 (even though there is a disagreement on the order). In addition, all the metrics include the set of $\{1, 0, 2, 18, 53, 54, 17, 56\}$ to be included in the vital set, while DINOv1 and DINOv2 suggest also including $\{28, 25\}$, CLIP suggests including $\{3, 4\}$ instead and LPIPS suggests including $\{6, 4\}$ instead. In Figure 12 we edited images with these slightly different vital layer sets, and found the differences to be negligible in practice.

2.3. VLM-Based Quantiative Metric

As explained in Section 4.1 of the main paper, We evaluated the editing results using three widely-used CLIP-based

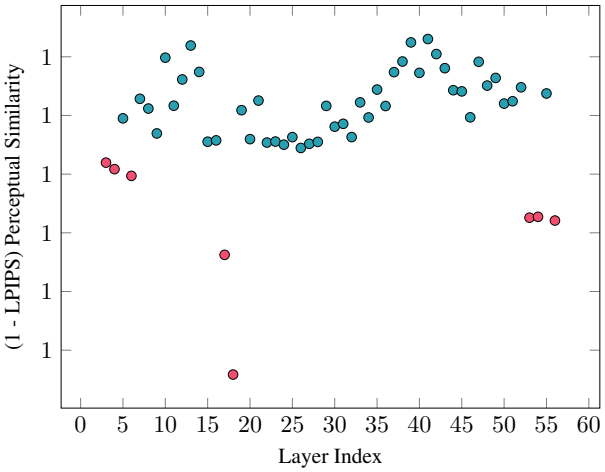


Figure 11. **Layer Removal Quantitative Comparison Using LPIPS.** As explained in Section 2.2, we measured the effect of removing each layer of the model by calculating the *(1 - LPIPS)* [24] perceptual similarity between the generated images with and without this layer. Lower perceptual similarity indicates significant changes in the generated images. As can be seen, removing certain layers significantly affects the generated images, while others have minimal impact. Importantly, influential layers are distributed across the transformer rather than concentrated in specific regions. Note that the first vital layers were omitted for clarity (as their perceptual similarity approached zero).

Table 2. **VLM-Based quantitative comparison.** For each method, we used Phi-3.5-vision [1] VLM to compute the percentage of editing results that follow the text prompt and of the results that change only the essential parts of the image. P2P+NTI [9, 14], Instruct-P2P [4], and MasaCTRL [5] suffer from low similarity to the text prompt. SDEdit [22] and MagicBrush [23] adhere more to the text prompt, but they struggle with avoiding unintended changes.

Method	Text Following (↑)	Modify only essential (↑)
SDEdit [88]	86.66%	21.66%
P2P+NTI [33, 51]	68.33%	26.66%
Instruct-P2P [17]	33.33%	26.66%
MagicBrush [91]	**88.33%**	46.66%
MasaCTRL [18]	33.33%	06.66%
Stable Flow (ours)	83.33%	**61.66%**

metrics: CLIP_{img}, CLIP_{txt}, and CLIP_{dir}. In addition, we experimented with a VLM-based metric using the Phi-3.5-vision [1] VLM that was trained specifically for the task of multiple image comparison. For each input image x, editing prompt p, and editing result $\hat{x}$, we computed the following two metrics: (1) *Text Following* — we presented the VLM the edited image $\hat{x}$ and the editing prompt p and asked it "Does this image correspond to the text p? Answer yes or no.". (2) *Modify only essential parts* — we extracted the prompt instruction and presented it, along with x and $\hat{x}$, to the VLM and asked it "Is the only difference between these two images the text PROMPT? Answer yes or no". For each metric, we calculated the number of times that the VLM answered "yes". As demonstrated in Table 2, the VLM-based metric follows the same trend as the CLIP-based metrics:P2P+NTI [9, 14], Instruct-P2P [4], and MasaCTRL [5] suffer from low similarity to the text prompt. SDEdit [22] and MagicBrush [23] adhere more to the text prompt, but they struggle with avoiding unintended changes.

2.4. Number of Vital Layers

The somewhat agnostic nature of our method to the specific perceptual metric, as described in Section 2.2, raises the question of the importance of the entire vital layer set V to the editing task. To this end, in Figure 13 we experimented with omitting a growing number of vital layers and testing the editing results. As can be seen, when removing 20% of the vital layer set, the changes are negligible. However, when removing more than that, the editing results include unintended changes, such as identity changes (*e.g.*, man and woman examples) and background changes (*e.g.*, cat and blackboard examples). This is consistent with the results from Section 2.2 that show that the least vital layers for each perceptual metric are less important for the image editing task.

2.5. Latent Nudging Experiments

As described in Section 3.3 of the main paper, we proposed using a latent nudging technique to avoid the bad reconstruction quality of vanilla inverse Euler ODE solver. We suggest multiplying the initial latent z_0 by a small scalar $\lambda = 1.15$ to slightly offset it from the training distribution. As shown in Figure 14, we empirically tested different values for the latent nudging hyperparameter λ. We performed inversion using the inverse Euler ODE solver with a high number of 1,000 inversion (and denoising) steps, to reduce the inversion error. However, even when using such a high number of inversion/denoising steps, we notice that when not using latent nudging (*i.e.*, $\lambda = 1.0$), the reconstruction quality is poor (notice the eyes and the legs of the dog). Next, we found that $\lambda = 1.15$ is the smallest value that enables full reconstruction using the inverse Euler solver. Furthermore, nudging values that are too high (*e.g.*, $\lambda = 3.0$) result in saturated images. Lastly, we notice that decreasing nudging values (*i.e.*, $\lambda < 1.0$) severely damages the reconstruction quality.

In addition, we experiment with a simpler inversion variant based on latent caching (termed *DDPM bucketing* in DiffUHaul [3]), in which we saved the series of latents during the inversion process without applying latent nudging. As shown in Figure 15, this approach indeed achieves perfect inversion (second column), but (third column) still struggles with preserving the identities while editing the im-

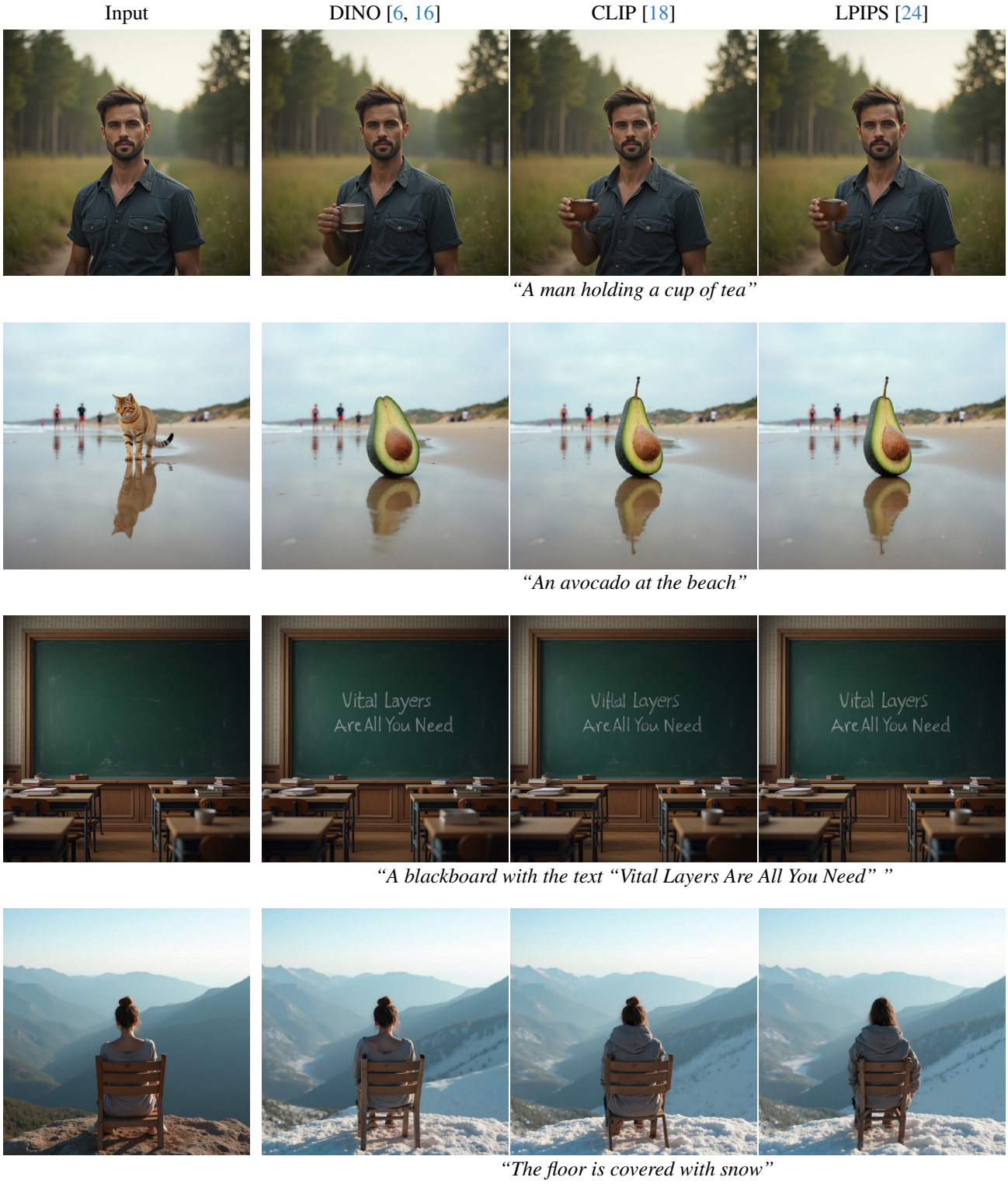


Figure 12. **Metrics Qualitative Comparison.** As described in Section 2.2, we also experimented with other perceptual metrics. We found DINOv2 [16] and DINOv1 [6] to produce the same set of vital layer. While CLIP [18] and LPIPS [24] replaced two layers in the vital layers set (though they include most of the vital layer set as in DINO). As can be seen, the differences between these sets are negligible when editing images.

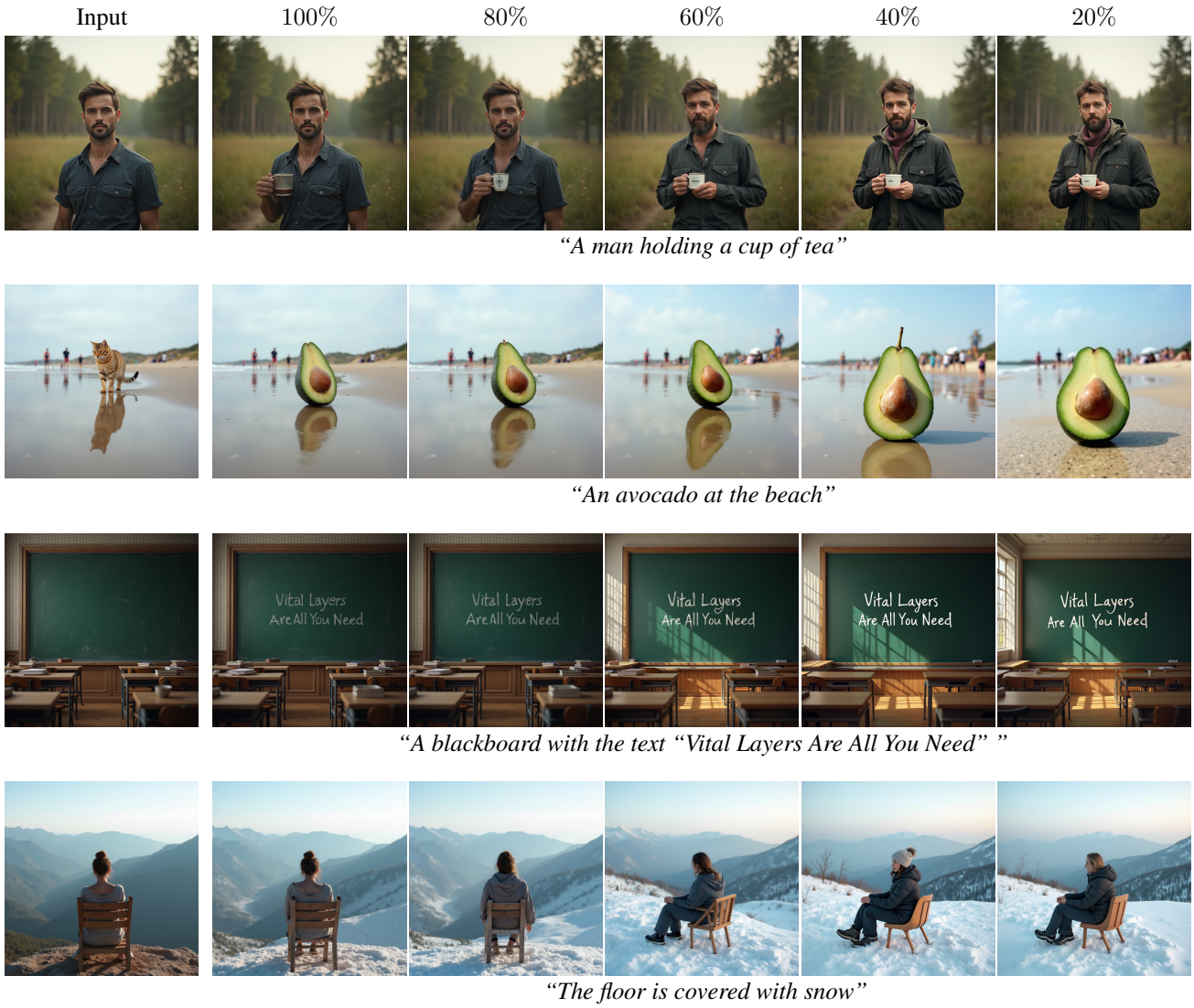


Figure 13. **Number of Vital Layers Comparison.** As explained in Section 2.4, we experimented with choosing a different portion of the calculated vital layer set V. As can be seen, when removing 20% of the vital layer set, the changes are negligible. However, when removing more than that, the editing results include unintended changes, such as identity changes (*e.g.*, the man and woman examples) and background changes (*e.g.*, the cat and blackboard examples).

age (*e.g.*, the rabbit and duck examples) or significantly alters the image (*e.g.*, the cat and man examples). On the other hand, our method (fourth column) with the latent nudging is able to preserve the identities during editing. In practice, we found that using latent caching in addition to latent nudging enables inversion with a lower number of steps (50 steps), hence, this is the approach we used.

2.6. Layer Bypassing Visualization

As explained in Section 3.1 of the main paper, to quantify layer importance in FLUX model, we devised a systematic evaluation approach. Using ChatGPT [15], we automatically generated a set P of $k = 64$ diverse text prompts, and draw a set S of random seeds. Each of these prompts was used to generate a reference image, yielding a set G_{ref}. For each DiT layer $\ell \in \mathbb{L}$, we performed an ablation by bypassing the layer using its residual connection. This process generated a set of images G_ℓ from the same prompts and seeds.

In Figures 16–23, we provide a full visualization of the reference set G_{ref} along with the generation sets $G_0 - G_{56}$. As can be seen, removing certain layers significantly affects the generated images, while others have minimal impact. Importantly, influential layers are distributed across the transformer rather than concentrated in specific regions.

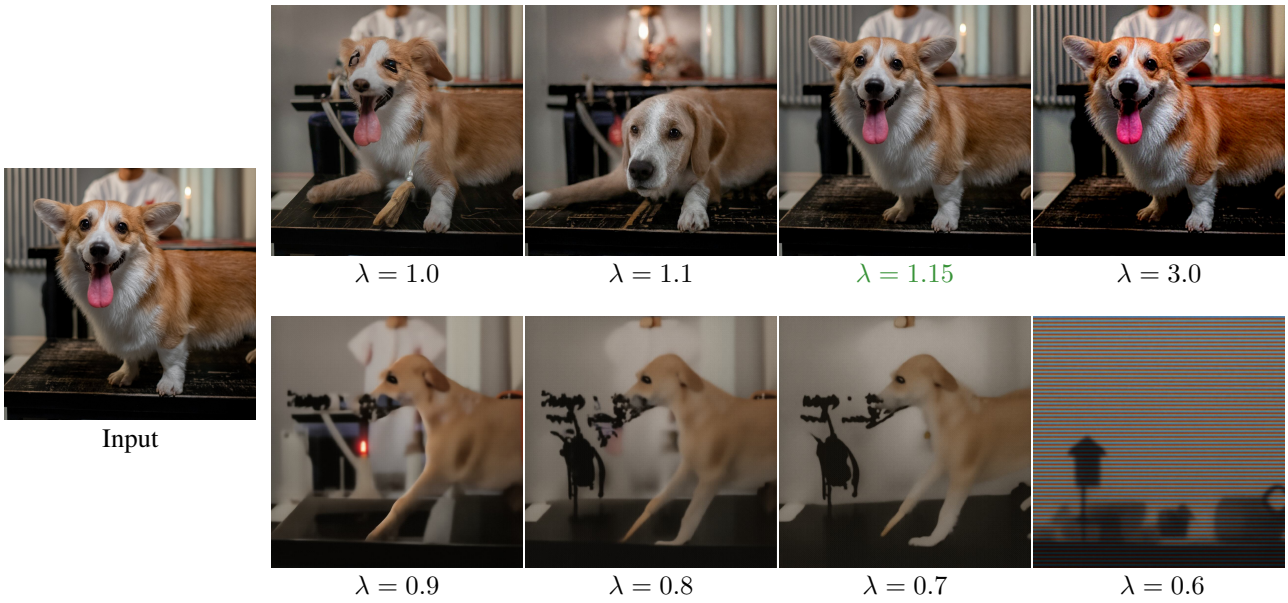


Figure 14. **Latent Nudging Values.** As described in Section 2.5, we empirically tested different values for the latent nudging hyperparameter λ. In our experiments, we performed inversion using the inverse Euler ODE solver with a high number of 1,000 inversion (and denoising) steps, to reduce the inversion error. However, even when using such a high number of inversion/denoising steps, we notice that when not using latent nudging (*i.e.*, $\lambda = 1.0$), the reconstruction quality is poor (notice the eyes and the legs of the dog). Next, we found that $\lambda = 1.15$ is the smallest value that enables full reconstruction using the inverse Euler solver. Furthermore, nudging values that are too high (*e.g.*, $\lambda = 3.0$) result in saturated images. Lastly, we notice that reducing nudging values ($\lambda < 1.0$) severely damages the reconstruction quality.

2.7. Stable Diffusion 3 Results

All the experiments in the main paper were based on the FLUX.1-dev [11] model. We also experimented with a different DiT text-to-image flow model named Stable Diffusion 3 [7] based on the Diffusers [19] implementation of the medium model.

As described in Section 3.1 of the main paper, we measured the importance of each of the layers of this model. As shown in Figure 24, we measured the effect of removing each layer from the model by calculating the perceptual similarity between the generated images with and without this layer. Lower perceptual similarity indicates significant changes in the generated images. As can be seen, removing certain layers significantly affects the generated images, while others have minimal impact.

Next, in Figure 25 we illustrate the qualitative differences between vital and non-vital layers. While bypassing non-vital layers (G_1 and G_{21}) results in modest alterations, bypassing vital layers leads to significant changes: complete noise generation (G_0) or severe distortions (G_7, G_8, and G_9).

Finally, in Figure 26, we perform various editing operations using the same mechanism of injecting the reference image information into the vital layers of the model, as described in Section 3.2 of the main paper.

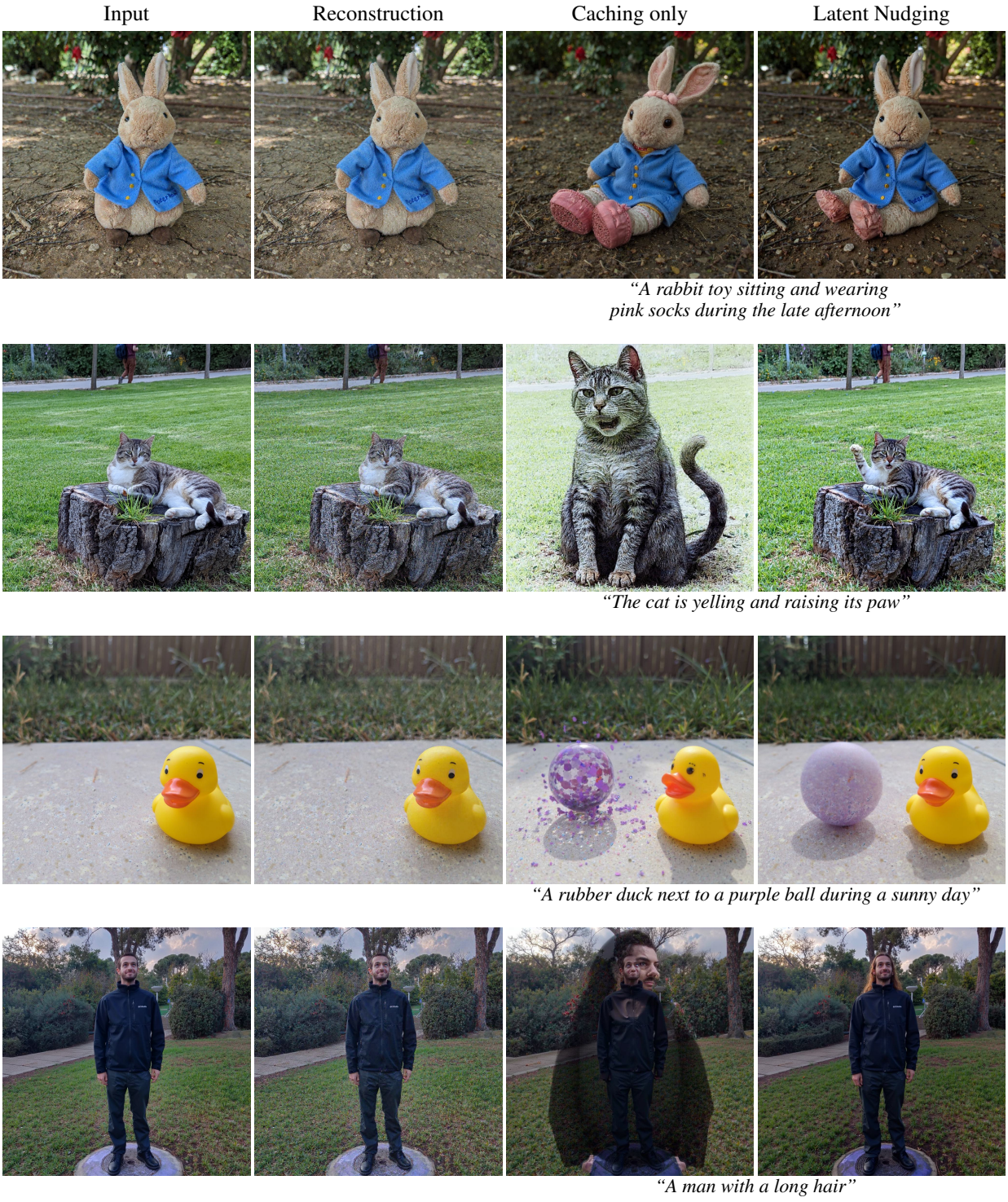


Figure 15. **Latent Caching.** As explained in Section 2.5, we also tested a latent caching approach [3], in which we saved the series of latents during the inversion process without applying latent nudging. As can be seen, this approach indeed achieves perfect inversion (second column), but (third column) still struggles with preserving the identities while editing the image (*e.g.*, the rabbit and duck examples) or significantly alters the image (*e.g.*, the cat and man examples). On the other hand, our method with the latent nudging (fourth column) is able to preserve the identities during editing.

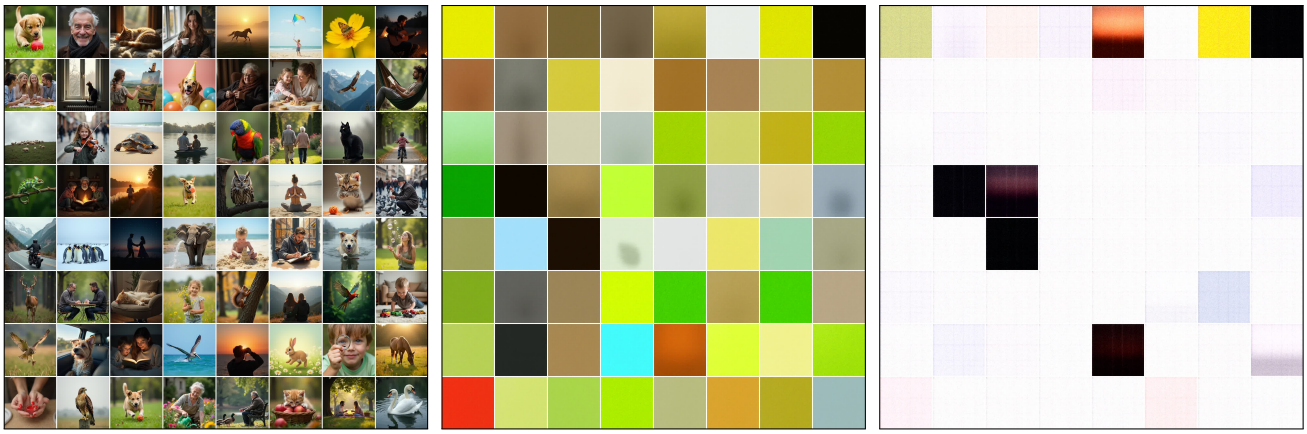


Figure 16. **Full Layer Bypassing Visualization for Flux.** We visualize the individual layer bypassing study we conducted, as described in Section 2.6. We start by generating a set of images G_{ref} using a fixed set of seeds and prompts. Then, we bypass each layer ℓ by using its residual connection and generate the set of images G_ℓ using the same fixed set of prompts and seeds. In this visualization, G_0 – G_2 are vital layers, while G_3 – G_7 are non-vital layers.

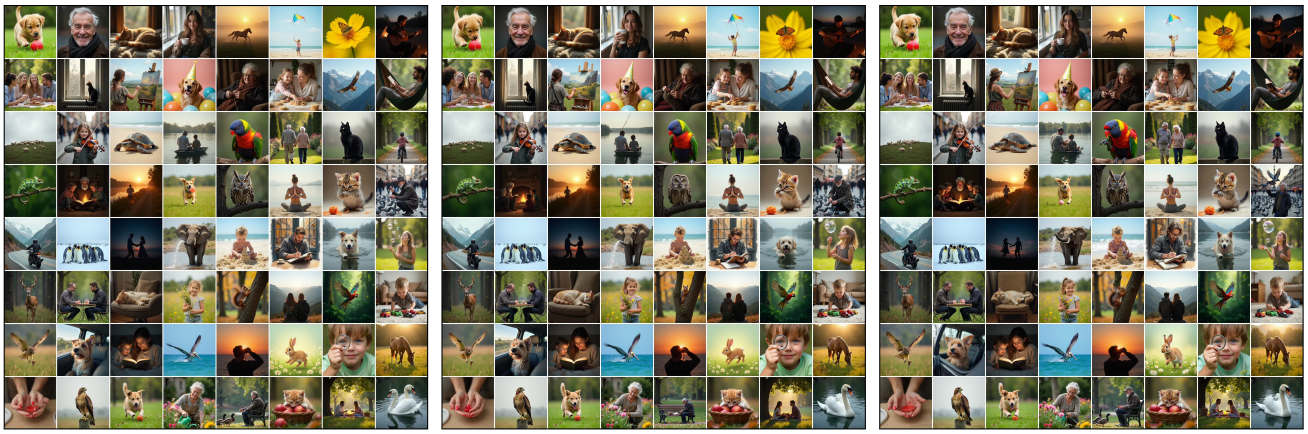
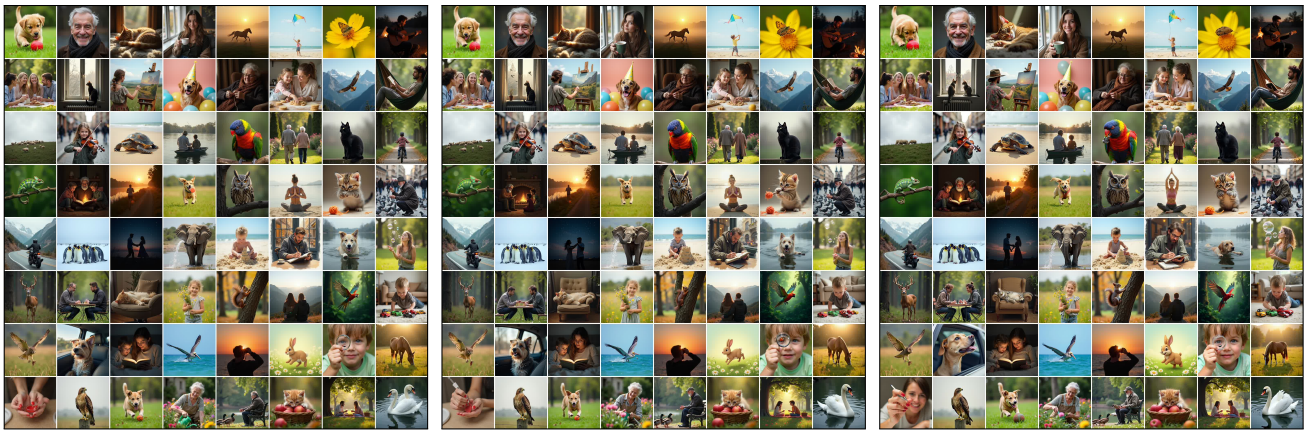
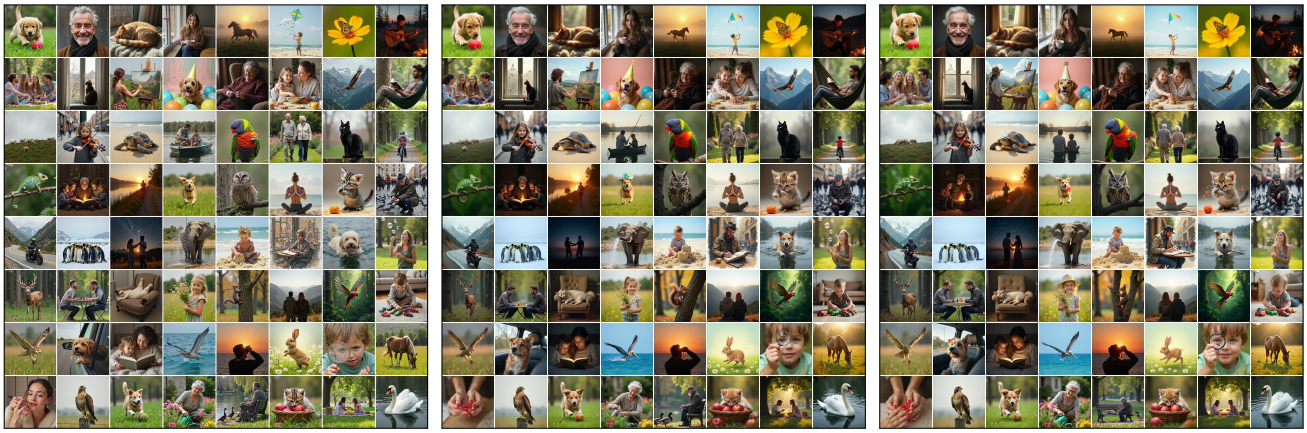


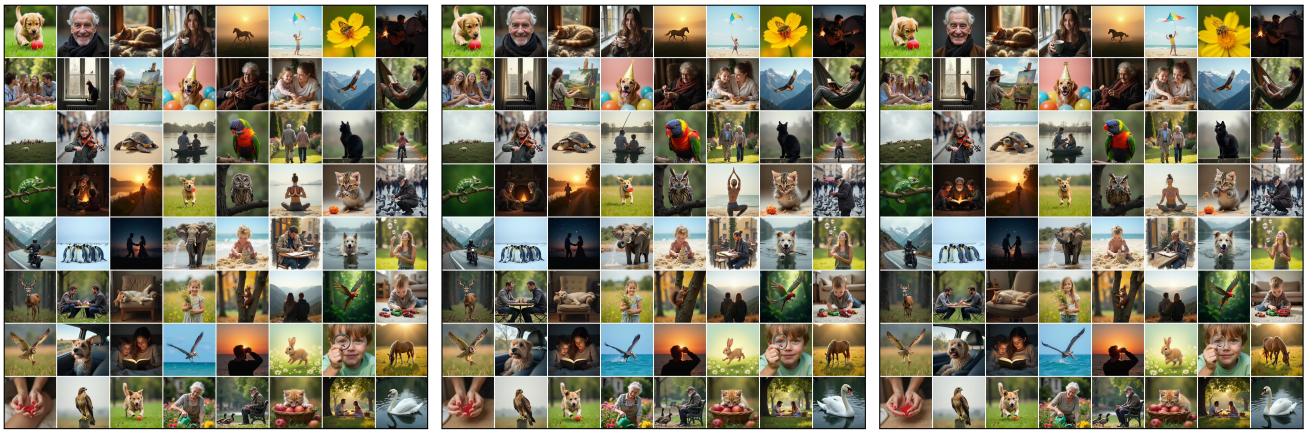
Figure 17. **Full Layer Bypassing Visualization for Flux.** We visualize the individual layer bypassing study we conducted, as described in Section 2.6. We start by generating a set of images G_{ref} using a fixed set of seeds and prompts. Then, we bypass each layer ℓ by using its residual connection and generate the set of images G_ℓ using the same fixed set of prompts and seeds. In this visualization, G_8 – G_{15} are all non-vital layers.



G_{ref} G_{16} G_{17}

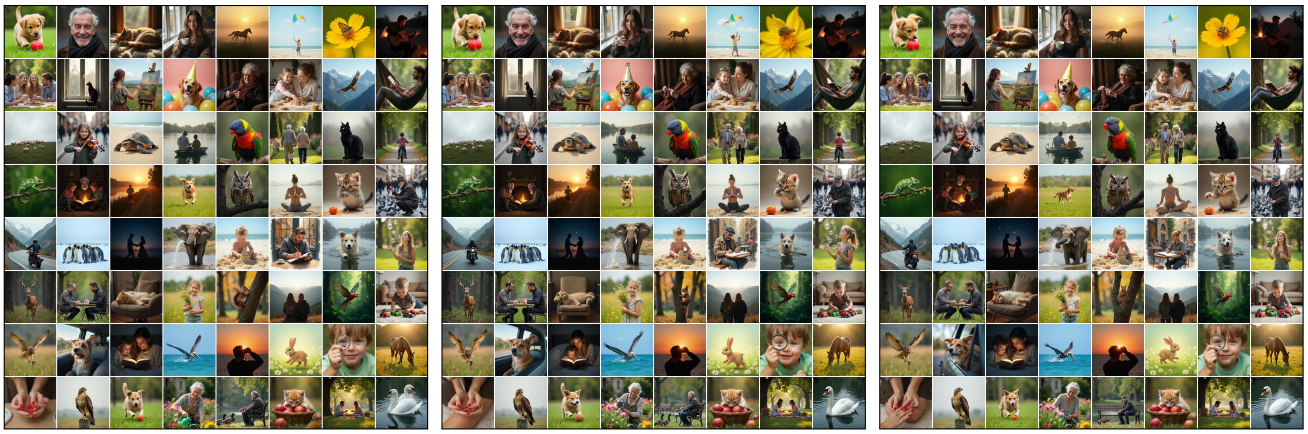


G_{18} G_{19} G_{20}

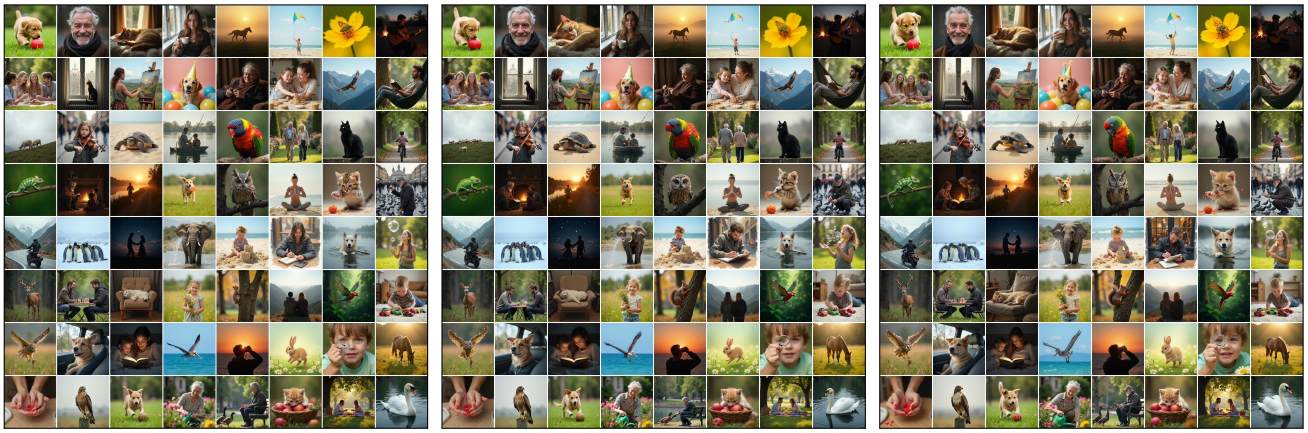


G_{21} G_{22} G_{23}

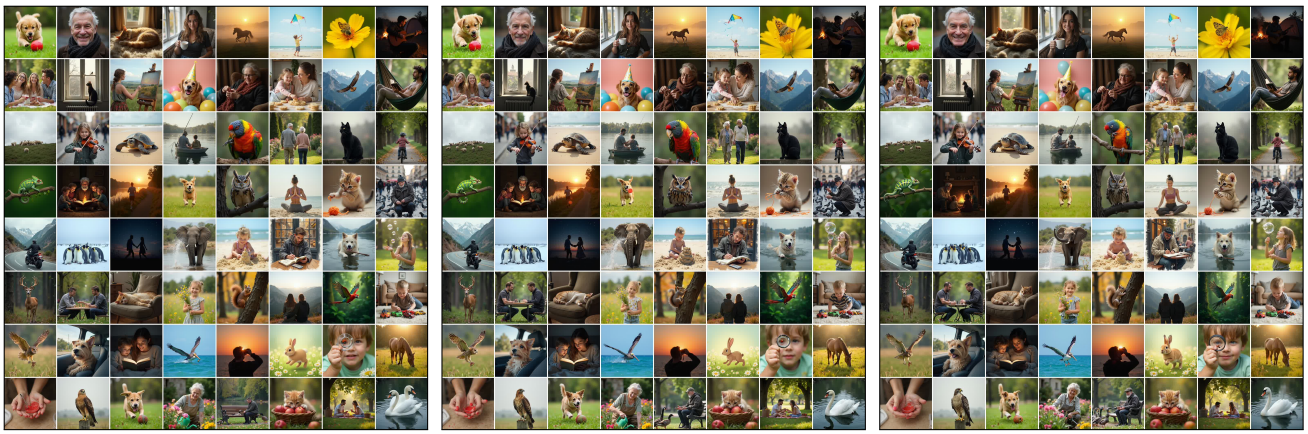
Figure 18. **Full Layer Bypassing Visualization for Flux.** We visualize the individual layer bypassing study we conducted, as described in Section 2.6. We start by generating a set of images G_{ref} using a fixed set of seeds and prompts. Then, we bypass each layer ℓ by using its residual connection and generate the set of images G_ℓ using the same fixed set of prompts and seeds. In this visualization, G_{17} and G_{18} are vital layers, while G_{16} and G_{19} – G_{23} are non-vital layers.



G_{ref} G_{24} G_{25}

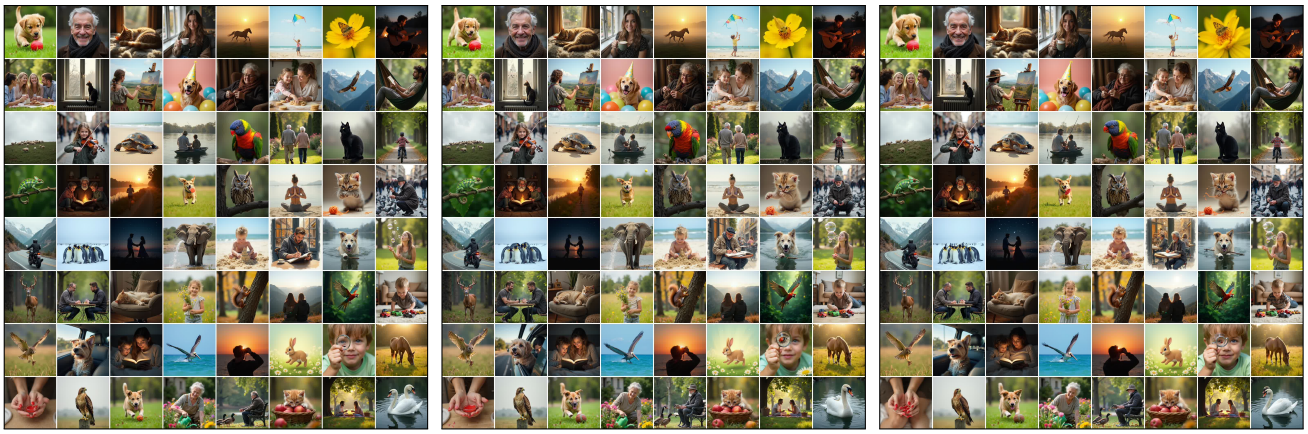


G_{26} G_{27} G_{28}

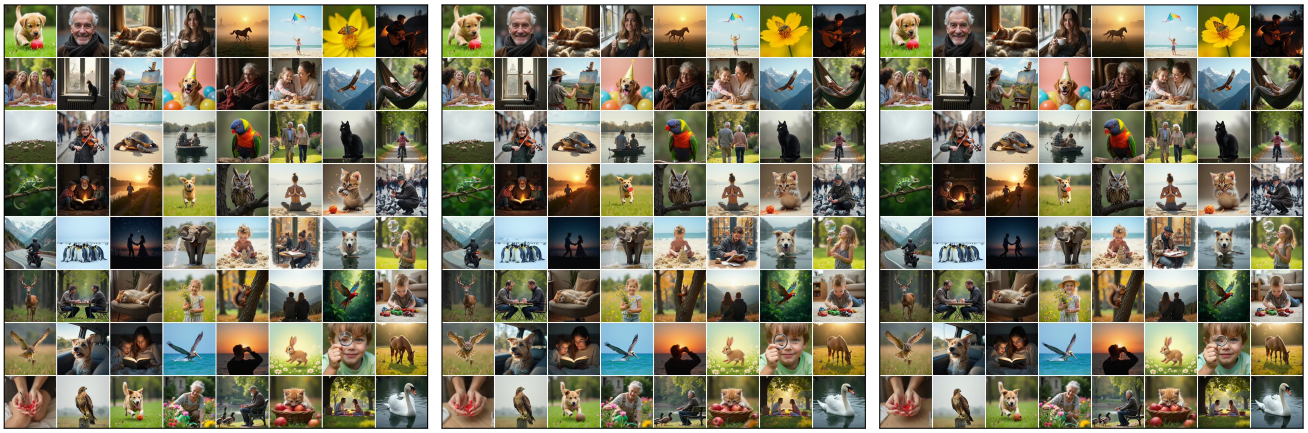


G_{29} G_{30} G_{31}

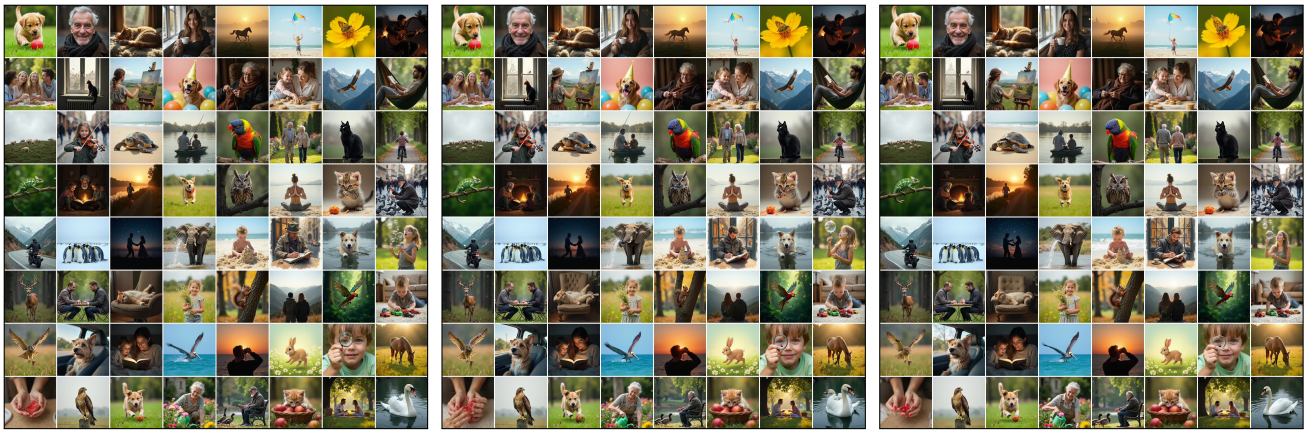
Figure 19. **Full Layer Bypassing Visualization for Flux.** We visualize the individual layer bypassing study we conducted, as described in Section 2.6. We start by generating a set of images G_{ref} using a fixed set of seeds and prompts. Then, we bypass each layer ℓ by using its residual connection and generate the set of images G_ℓ using the same fixed set of prompts and seeds. In this visualization, G_{25} and G_{28} are vital layers, while G_{24}, G_{26} – G_{27} and G_{29} – G_{31} are non-vital layers.



G_{ref} G_{32} G_{33}

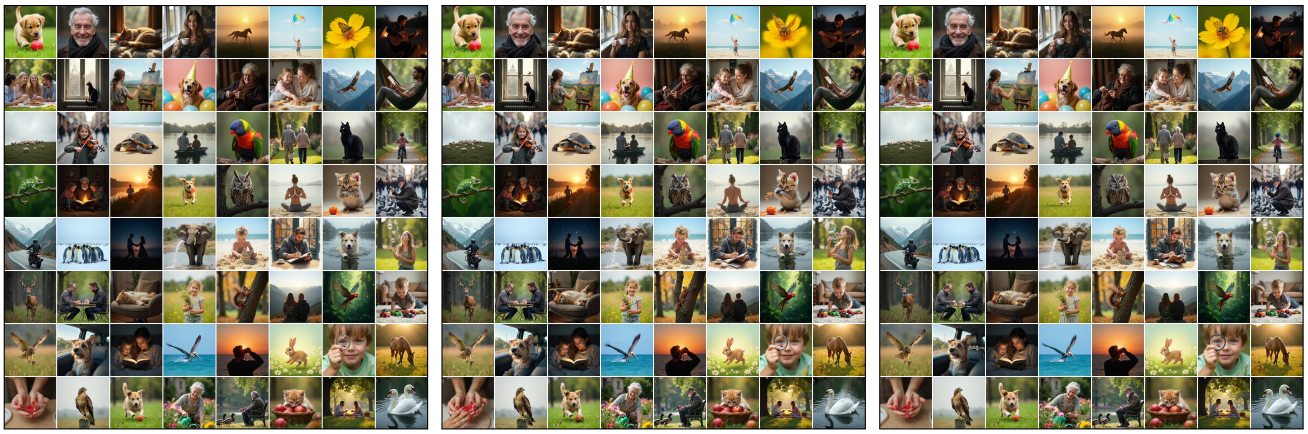


G_{34} G_{35} G_{36}

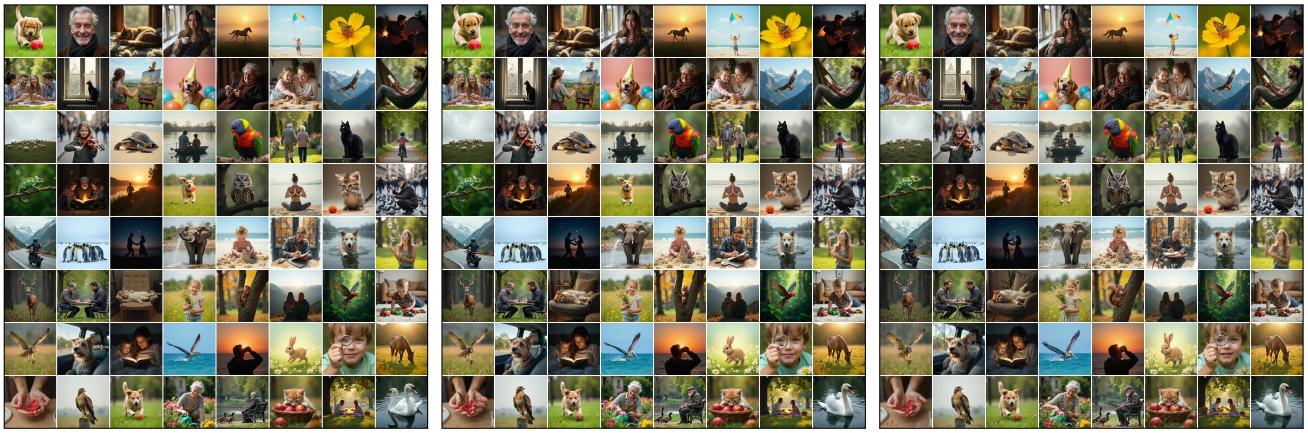


G_{37} G_{38} G_{39}

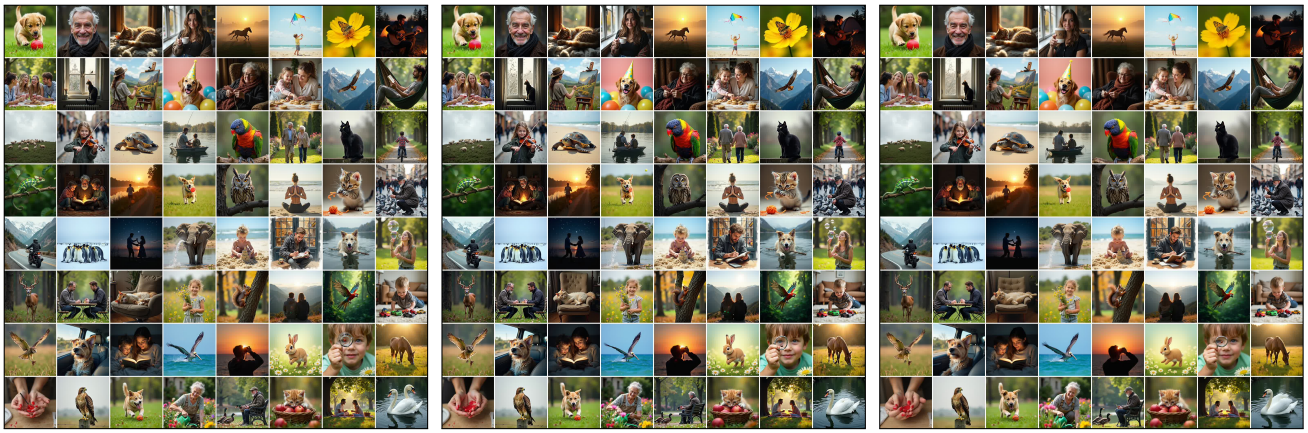
Figure 20. **Full Layer Bypassing Visualization for Flux.** We visualize the individual layer bypassing study we conducted, as described in Section 2.6. We start by generating a set of images G_{ref} using a fixed set of seeds and prompts. Then, we bypass each layer ℓ by using its residual connection and generate the set of images G_ℓ using the same fixed set of prompts and seeds. In this visualization, G_{31} – G_{39} are non-vital layers.



G_{ref} G_{40} G_{41}

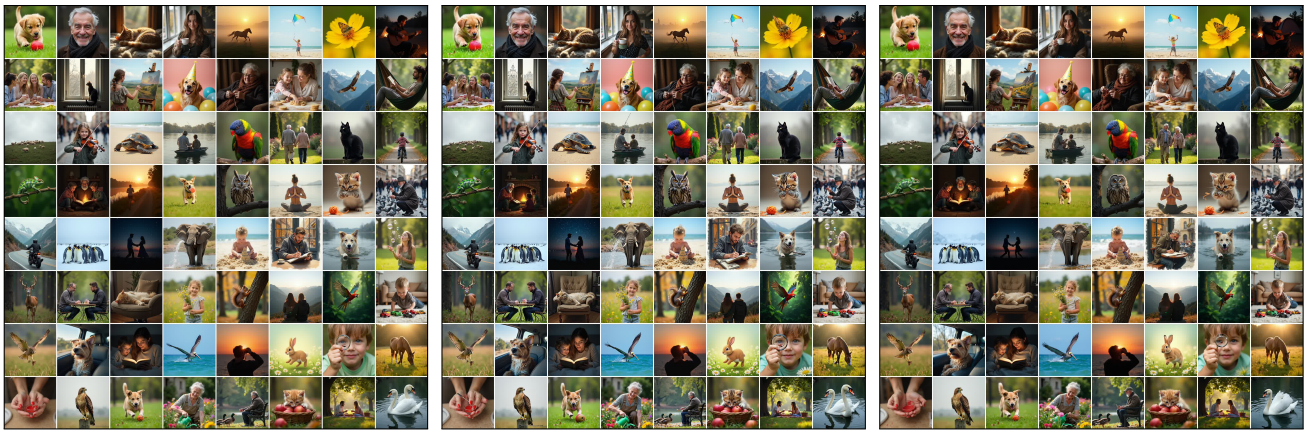


G_{42} G_{43} G_{44}

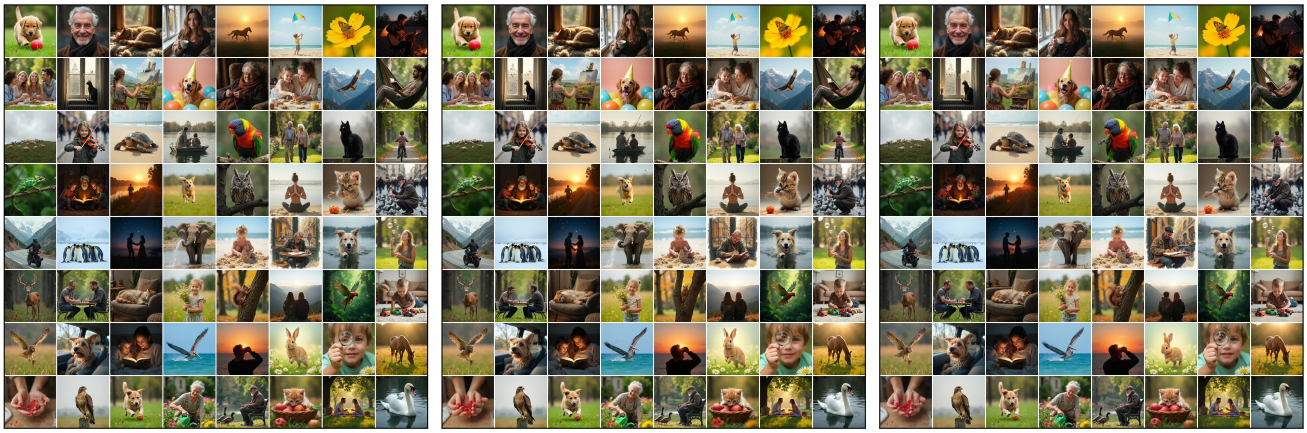


G_{45} G_{46} G_{47}

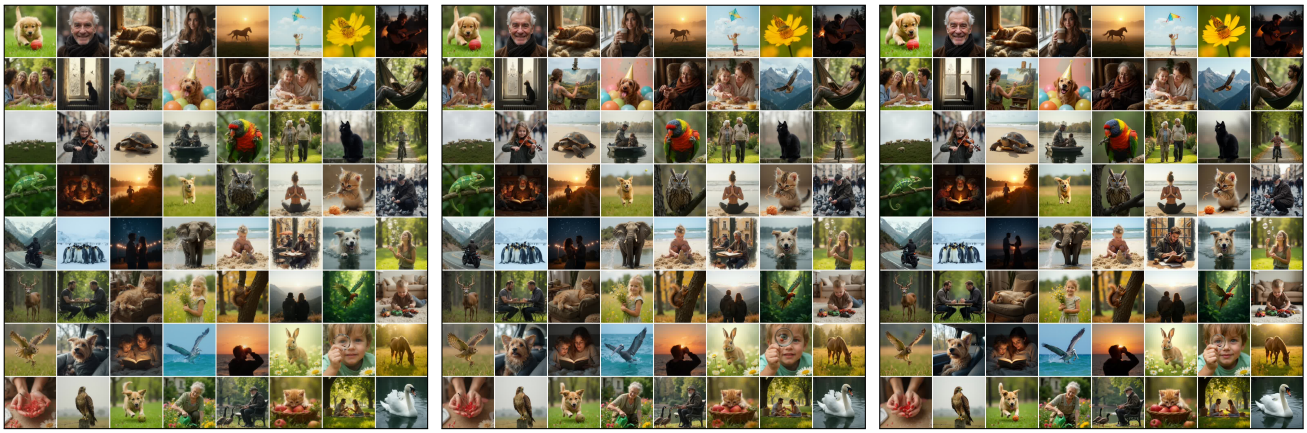
Figure 21. **Full Layer Bypassing Visualization for Flux.** We visualize the individual layer bypassing study we conducted, as described in Section 2.6. We start by generating a set of images G_{ref} using a fixed set of seeds and prompts. Then, we bypass each layer ℓ by using its residual connection and generate the set of images G_ℓ using the same fixed set of prompts and seeds. In this visualization, G_{40} – G_{47} are non-vital layers.



G_{ref} G_{48} G_{49}



G_{50} G_{51} G_{52}



G_{53} G_{54} G_{55}

Figure 22. **Full Layer Bypassing Visualization for Flux.** We visualize the individual layer bypassing study we conducted, as described in Section 2.6. We start by generating a set of images G_{ref} using a fixed set of seeds and prompts. Then, we bypass each layer ℓ by using its residual connection and generate the set of images G_ℓ using the same fixed set of prompts and seeds. In this visualization, G_{53} – G_{54} are vital layers, while G_{48} – G_{52} and G_{55} are non-vital layers.

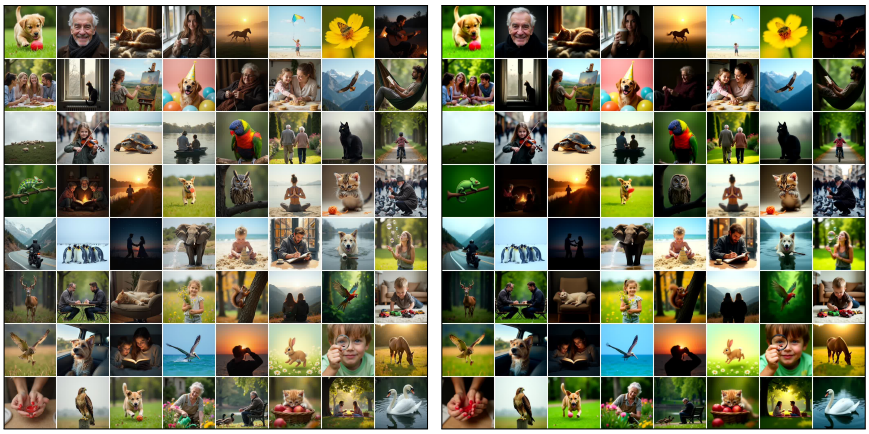


Figure 23. **Full Layer Bypassing Visualization for Flux.** We visualize the individual layer bypassing study we conducted, as described in Section 2.6. We start by generating a set of images G_{ref} using a fixed set of seeds and prompts. Then, we bypass each layer ℓ by using its residual connection and generate the set of images G_ℓ using the same fixed set of prompts and seeds. In this visualization, G_{56} is a vital layer.

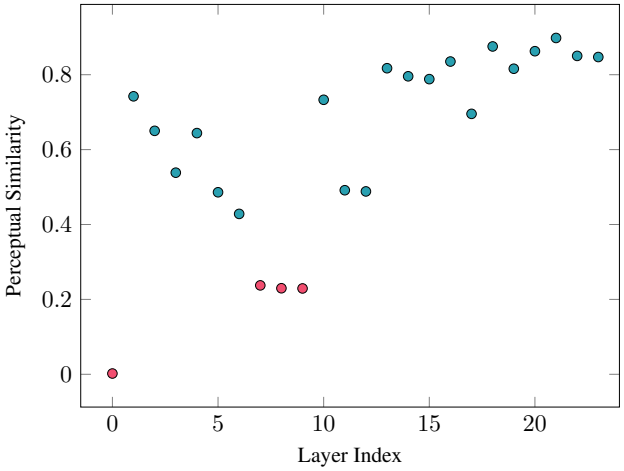


Figure 24. **Layer Removal Quantitative Comparison Stable Diffusion 3.** As explained in Section 2.7, we measured the effect of removing each layer of the model by calculating the perceptual similarity between the generated images with and without this layer. Lower perceptual similarity indicates significant changes in the generated images. As can be seen, removing certain layers significantly affects the generated images, while others have minimal impact. For a visual comparison, please refer to Figure 25.

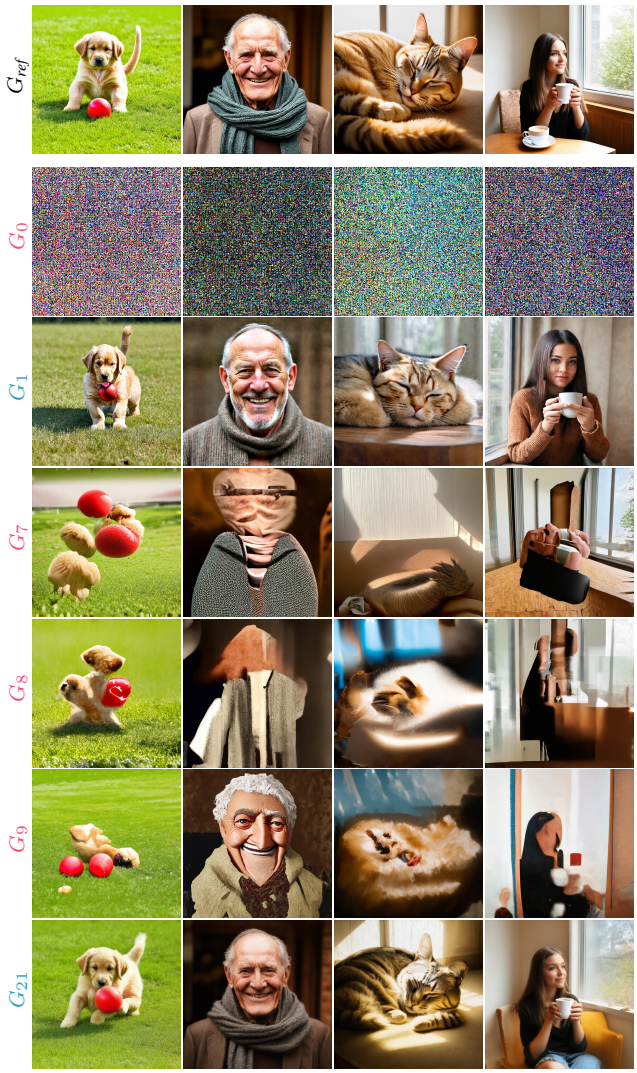


Figure 25. **Layer Removal Qualitative Comparison Stable Diffusion 3.** As explained in Section 2.7, we illustrate the qualitative differences between vital and non-vital layers. While bypassing non-vital layers (G_1 and G_{21}) results in modest alterations, bypassing vital layers leads to significant changes: complete noise generation (G_0), or severe distortions (G_7, G_8 and G_9). For a quantitative comparison, please refer to Figure 24

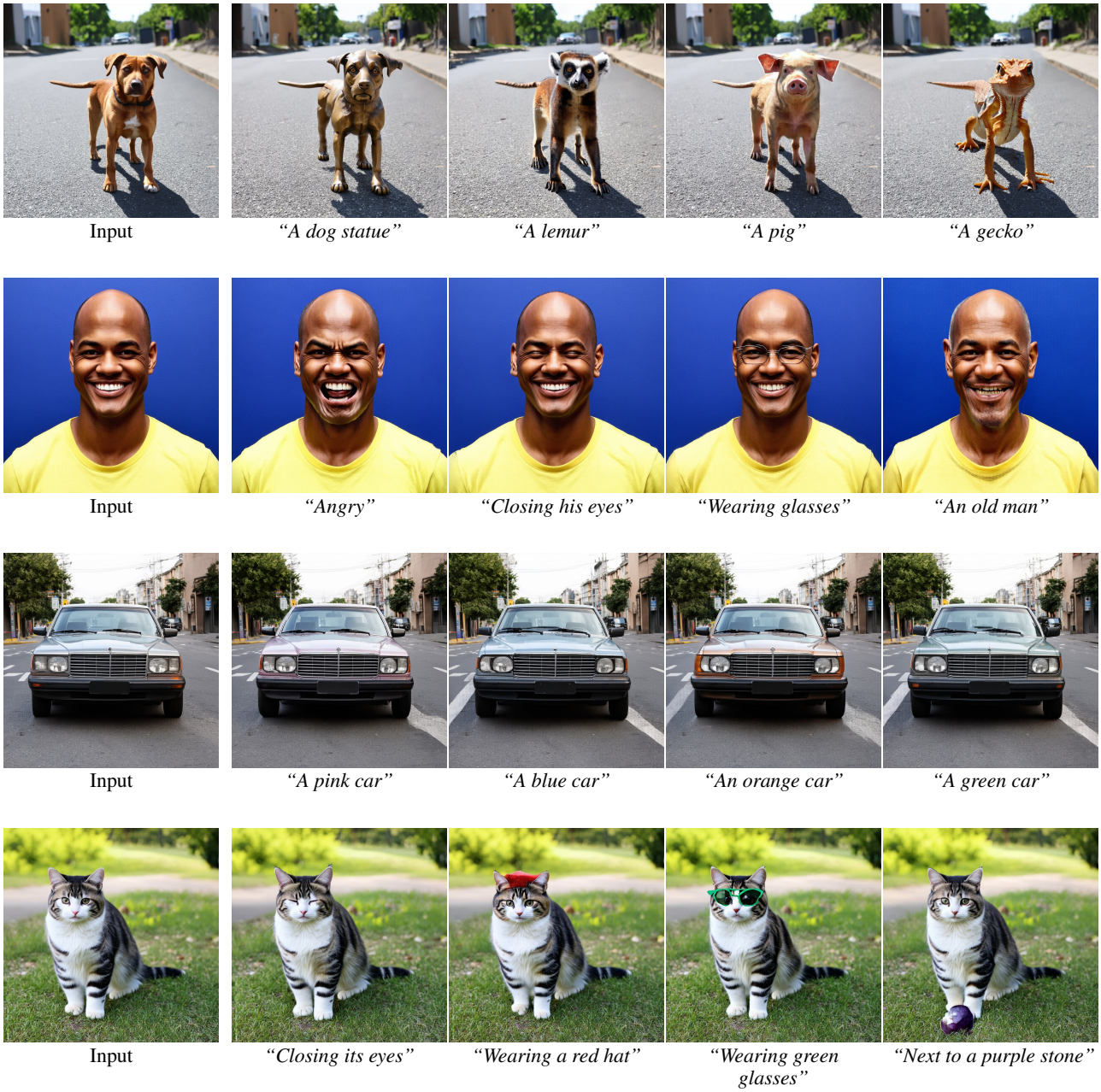


Figure 26. **Stable Diffusion 3 Editing Results.** As explained in Section 2.7, we tested our Stable Flow method on the Stable Diffusion 3 backbone [7]. As can be seen, we are able to perform various editing operations using the same mechanism of injecting the reference image information into the vital layers of the model.

References

[1] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Hassan Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Singh Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio C'esar Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allison Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Young Jin Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xianmin Song, Olatunji Ruwase, Praneetha Vaddamanu, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Cheng-Yuan Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone. *ArXiv*, abs/2404.14219, 2024. 10

[2] Amazon. Amazon mechanical turk. https://www.mturk.com/, 2024. 2, 3

[3] Omri Avrahami, Rinon Gal, Gal Chechik, Ohad Fried, Dani Lischinski, Arash Vahdat, and Weili Nie. Diffuhaul: A training-free method for object dragging in images. *arXiv preprint arXiv:2406.01594*, 2024. 10, 14

[4] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 1, 2, 4, 9, 10

[5] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22560–22570, 2023. 1, 2, 4, 9, 10

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021. 2, 9, 11

[7] Patrick Esser, Sumith Kulal, A. Blattmann, Rahim Entezari, Jonas Muller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. *ArXiv*, abs/2403.03206, 2024. 1, 13, 24

[8] Rinon Gal, Or Patashnik, Haggai Maron, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada. *ACM Transactions on Graphics (TOG)*, 41:1 – 13, 2021. 2

[9] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-prompt image editing with cross-attention control. In *The Eleventh International Conference on Learning Representations*, 2022. 1, 2, 4, 9, 10

[10] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *ArXiv*, abs/1603.08155, 2016. 9

[11] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 1, 13

[12] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 2, 3, 4, 5

[13] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2021. 1, 2, 3, 4

[14] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 1, 2, 4, 9, 10

[15] OpenAI. ChatGPT. https://chat.openai.com/, 2022. Accessed: 2024-10-1. 1, 12

[16] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Q. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russ Howes, Po-Yao (Bernie) Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Huijiao Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *ArXiv*, abs/2304.07193, 2023. 1, 2, 9, 11

[17] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2065–2074, 2021. 2

[18] 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 *International Conference on Machine Learning*, 2021. 2, 9, 11

[19] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 1, 13

[20] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama

Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, 2020. Association for Computational Linguistics. 2

[21] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Weixian Lei, Yuchao Gu, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7589–7599, 2022. 1

[22] 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. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18381–18391, 2022. 10

[23] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In *Advances in Neural Information Processing Systems*, 2023. 1, 2, 4, 9, 10

[24] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 2, 9, 10, 11