Stable Flow: Vital Layers for Training-Free Image Editing

CVPR 2025

Omri Avrahami Or Patashnik Ohad Fried Egor Nemchinov
Kfir Aberman Dani Lischinski Daniel Cohen-Or
Snap Research The Hebrew University of Jerusalem Tel Aviv University Reichman University

&

FreeFlux: Understanding and Exploiting Layer-Specific Roles in RoPE-Based MMDiTforVersatile Image Editing

ICCV 2025

Tianyi Wei Yifan Zhou Dongdong Chen Xingang Pan
S-Lab Nanyang Technological University Microsoft GenAI

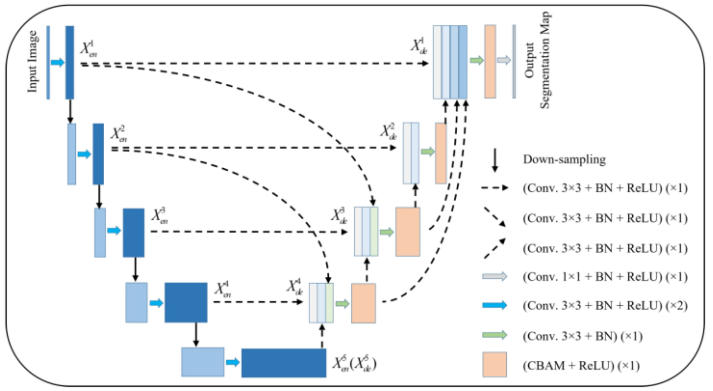
STRUCT Group Seminar
Presenter: Lan Xicheng
2026.01.19

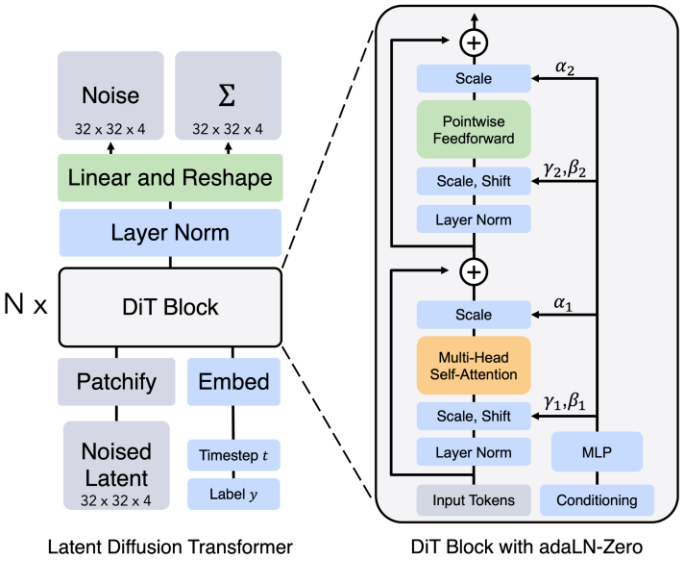
Outline

- Authors
- Background
- Stable Flow: Vital Layers Analysis
- FreeFlux: RoPE-Based Mechanism Analysis
- Experiments
- Conclusion

Background – UNet vs. DiT

- UNet Architecture
  - Distinct Downsampling / Upsampling blocks
  - Clear separation of semantic levels
  - Editing intuition: Manipulate specific decoder layers.
- DiT Architecture:
  - Repeated architecture
  - No explicit coarse-to-fine structure
  - Challenge: Where should we inject features for editing?





Latent Diffusion Transformer

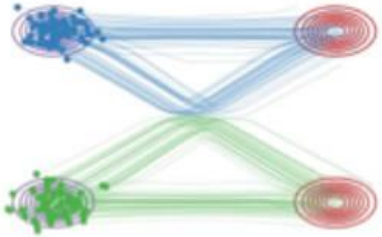
DiT Block with adaLN-Zero

Background – Flow Matching vs. Diffusion

- Diffusion: Models curved stochastic paths (SDE)
- Flow Matching: Models straight velocity fields (ODE)
- Faster sampling, better couplings

$$\min_v \int_0^1 \mathbb{E}\left[\|(X_1 - X_0) - v(X_t, t)\|^2\right] \mathrm{d}t. \quad X_t = tX_1 + (1-t)X_0, \quad t \in [0,1].$$

Linear interpolations



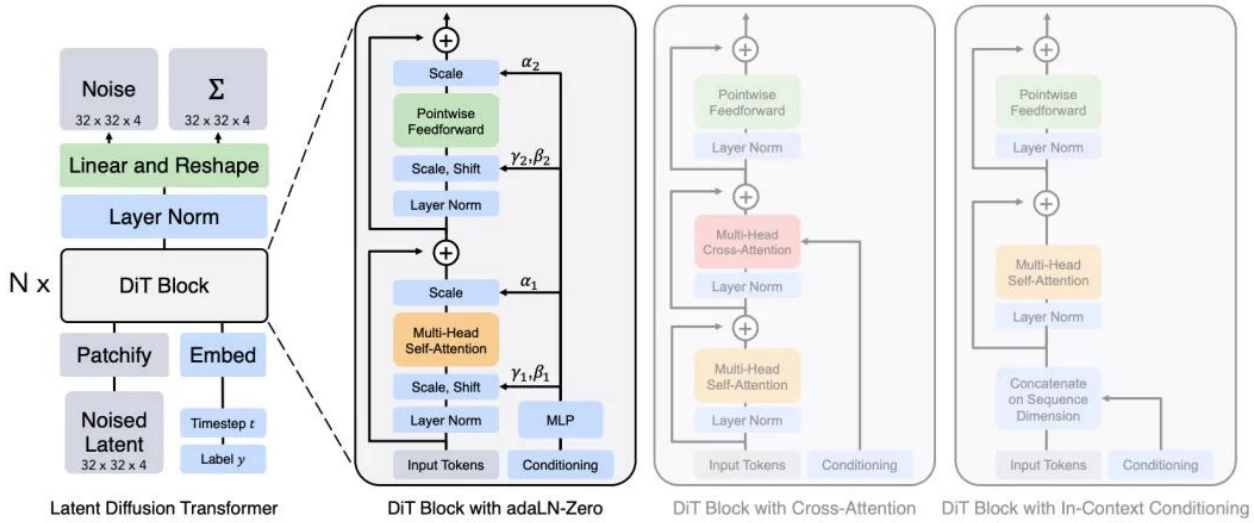
Learned vector field

Background – Flow Matching vs. Diffusion

- SDXL vs. Flux:
  - SDXL: Different prompts -> Different layouts/poses
  - Flux: Different prompts -> Same layout, only semantic changes

Background – Diffusion Transformer

- Patchify: Image → Latent → Patches → Linear Embeddings
- Tokens: Image is treated as a sequence of tokens, similar to NLP
- Self-Attention: Global context interaction at every layer
- Scalability: Performance scales well with compute (FLOPs) and parameters

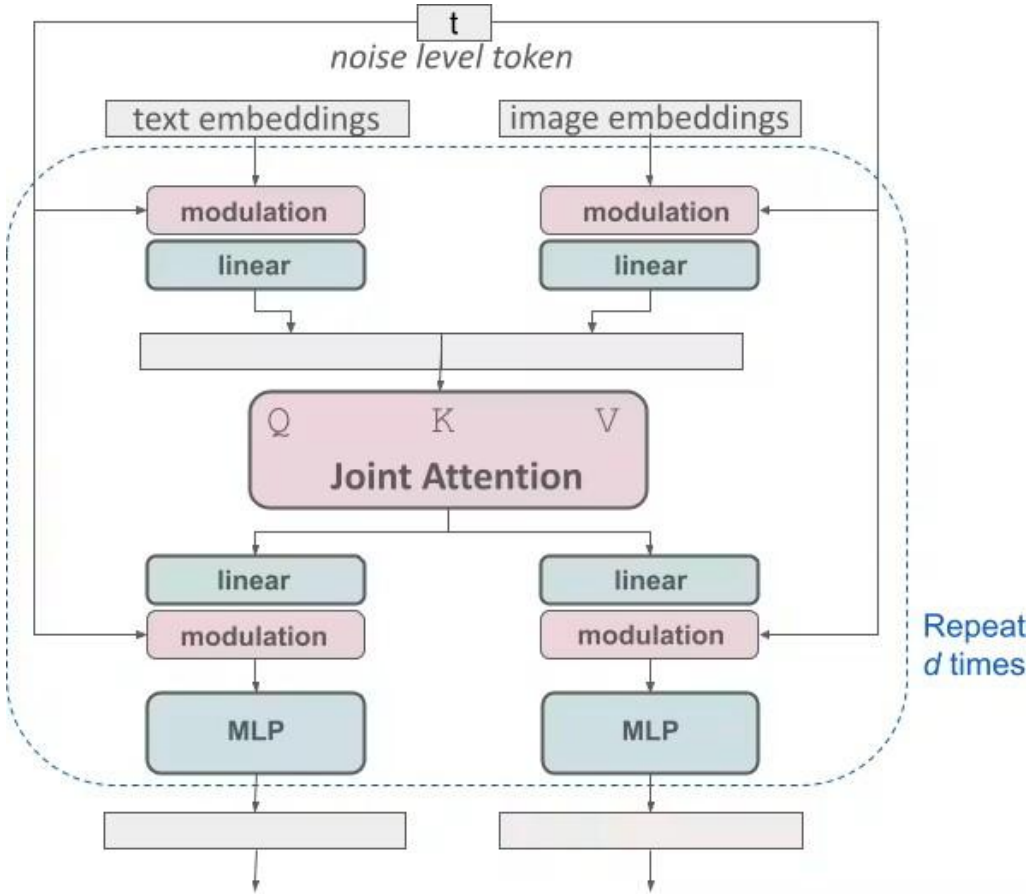


Background – Flux Architecture

- Flow Matching Backbone: Rectified Flow for straighter generation paths
- Structure:
  - Double Blocks (Multi-Stream): Separate processing for Text/Image, followed by joint attention. (19 blocks in Flux.1-dev)
  - Single Blocks (Single-Stream): Concatenated stream sharing same weights. (38 blocks in Flux.1-dev)
- Huge Parameter Count: ~12B parameters

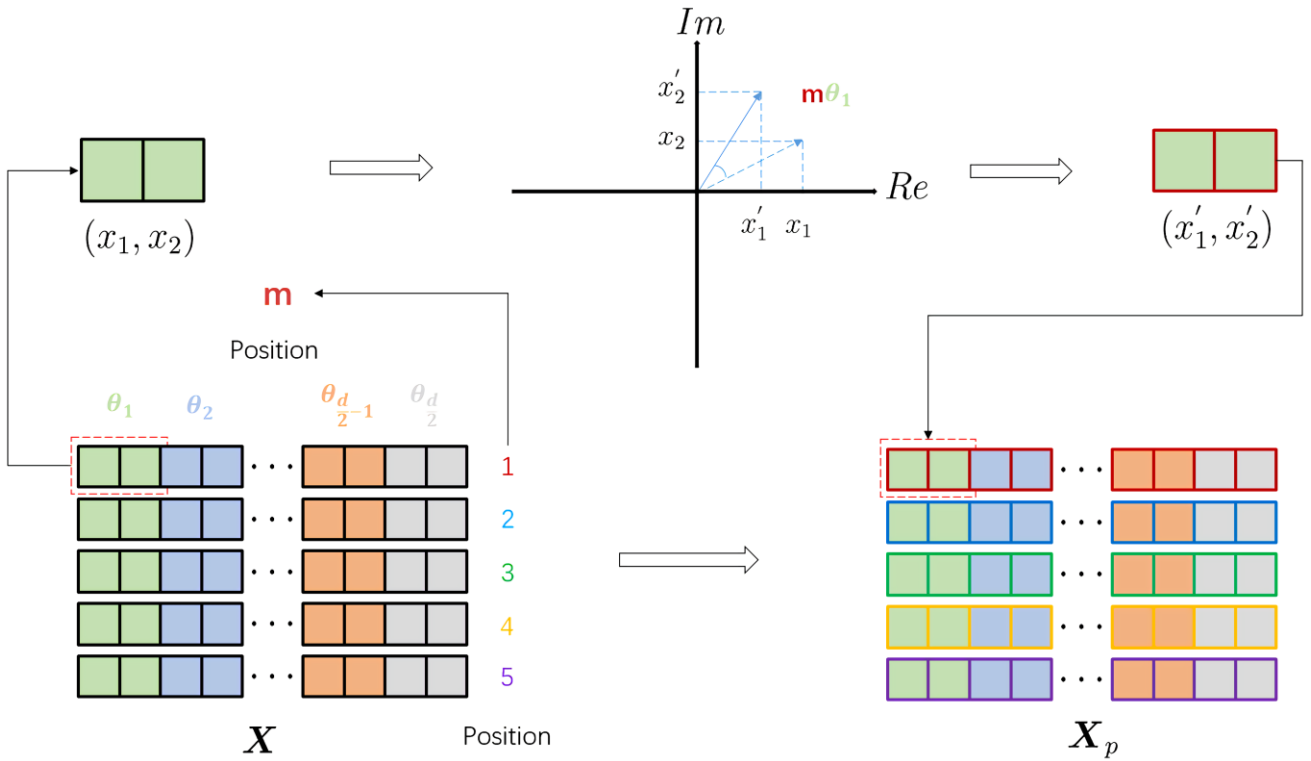
Background – Multimodal Diffusion Transformer

- Separate Weights: Text and Image have separate embedding weights
- Joint Attention: Text tokens and Image tokens interact in a unified attention matrix
- Bidirectional: Image attends to Text, Text attends to Image



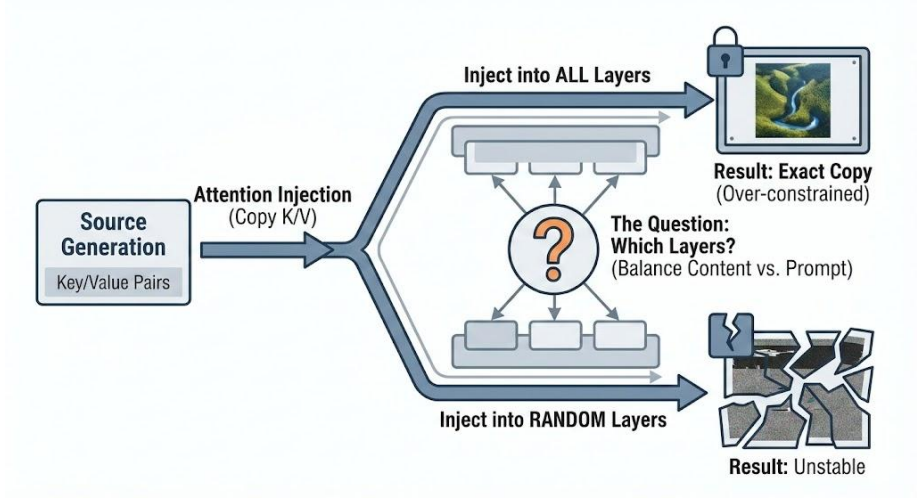
Background – Rotary Positional Embeddings

- Rotary Positional Embeddings (RoPE)
  - Mechanism: Encodes position by rotating query/key vectors
  - Feature: Captures relative positions effectively
  - In Flux: Applied to both Queries (Q) and Keys (K) at every layer



Background – Editing Challenge in DiT

- Attention Injection: A standard technique for training-free editing
- Copy Key/Value from Source generation to Target generation
- The Problem:
  - Injecting into all layers -> Over-constrains (Copies source image exactly)
  - Injecting into random layers -> Unstable results
  - Which layers are responsible for preserving content vs. following prompts?



Background – Inversion in Flow Models

- Inversion: Mapping a real image back to its initial noise z_t
- Euler Inversion: Simply reversing the ODE step
- Problem: Simple inversion in Flux often fails to reconstruct the image perfectly or leads to artifacts during editing

Motivation - The "Black Box" Dilemma

- Structural Ambiguity
  - UNet: Explicit Coarse-to-Fine
  - DiT: Isotropic Architecture

- Research Goal
  - Decipher the Layers: Find which specific layers control Structure vs. Appearance

Outline

- Authors
- Background
- Stable Flow: Vital Layers Analysis
- FreeFlux: RoPE-Based Mechanism Analysis
- Experiments
- Conclusion

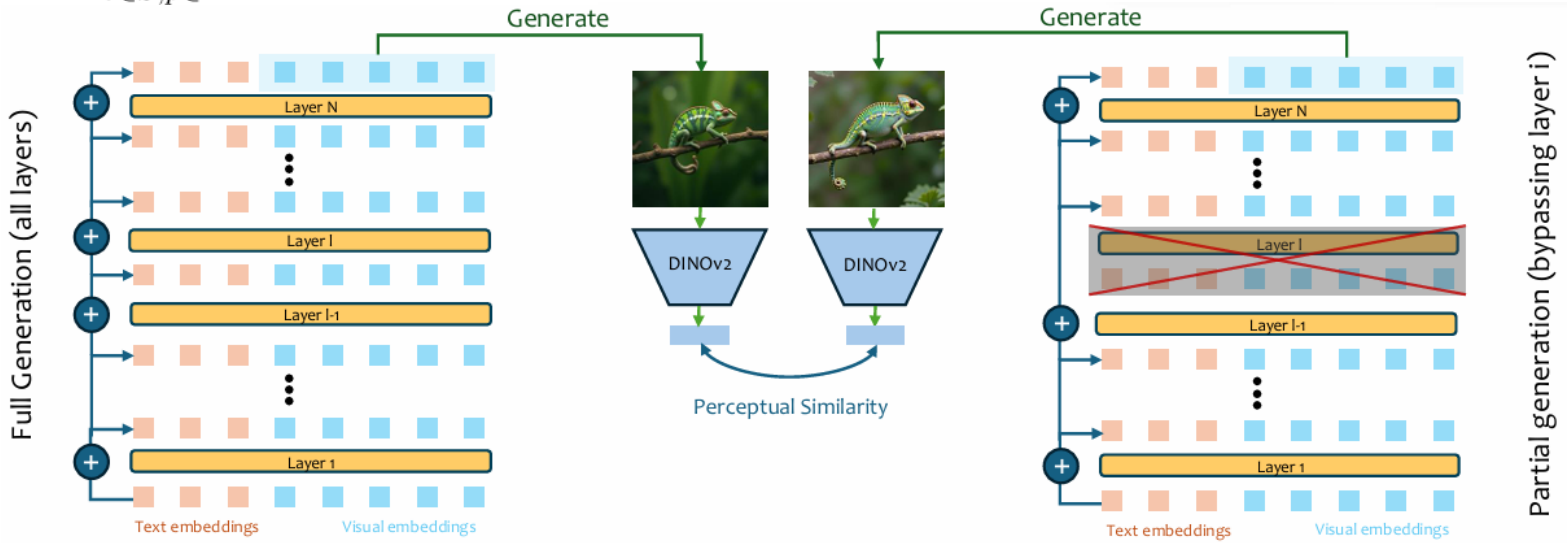
Methods - Stable Flow

- Phenomenon: Changing the prompt in Flux results in highly aligned structures
- Opportunity: This natural stability simplifies the editing task

Methods - Stable Flow

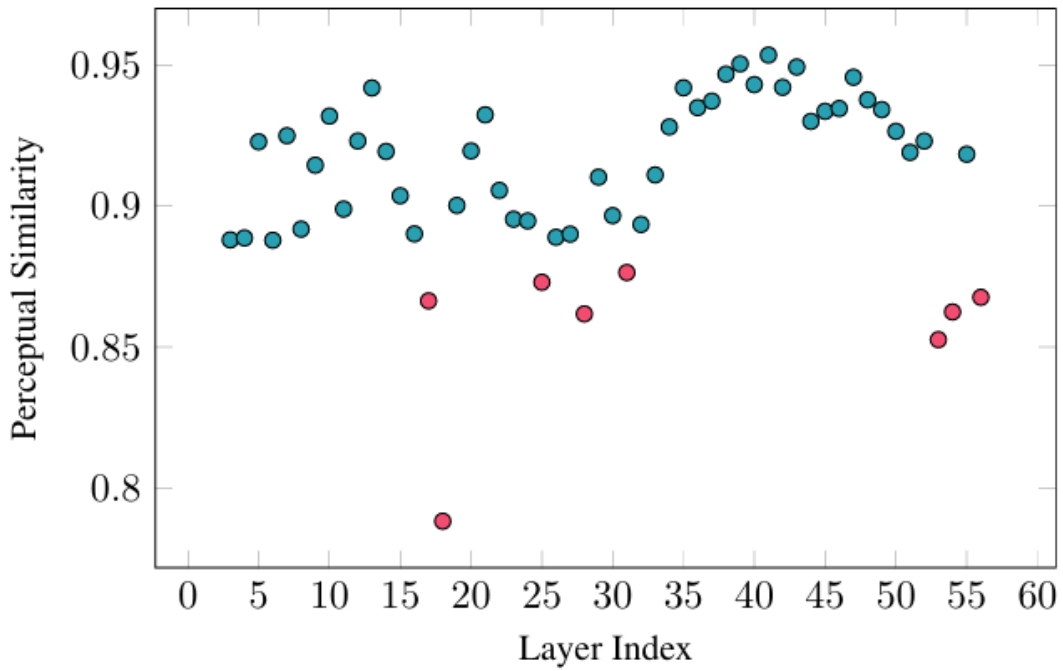
- If a layer is important, skipping it should drastically change the output
- Experiment
  - Generate reference image with full model
  - Generate ablated image by skipping layer
  - Measure Perceptual Similarity (DINOv2)

$$vitality(\ell) = 1 - \frac{1}{k} \sum_{s \in S, p \in P} d(M_{\text{full}}(s,p), M_{\text{-}\ell}(s,p))$$



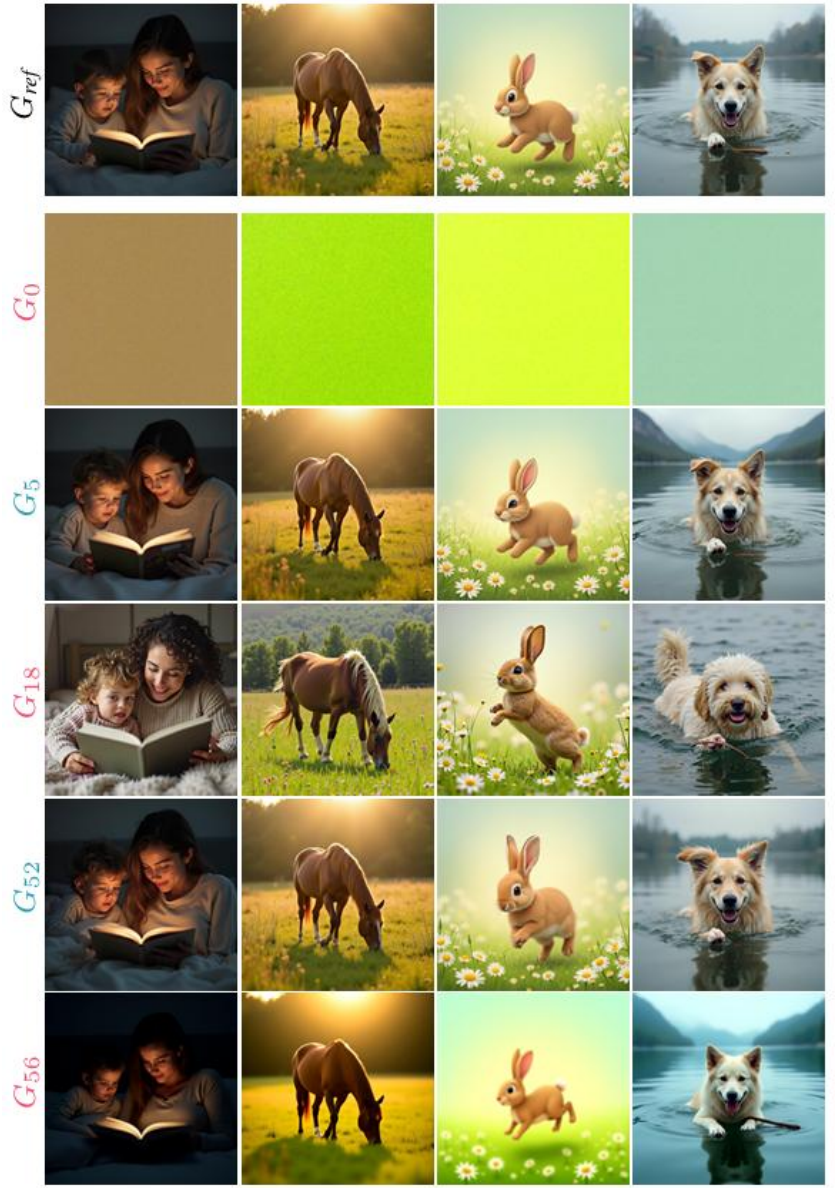
Methods - Stable Flow

- Findings
  - Vital layers are sparse
  - Distributed throughout the network (not just early or late)
  - Specific indices for Flux: e.g., Layers 0, 1, 17, 18, 53, 54... (scattered)



Methods - Stable Flow

- Skipping Non-Vital Layers: Minimal change to image
- Skipping Vital Layers: Complete noise, structural collapse, or identity loss
- Conclusion: Editing should focus on injecting features into Vital Layers

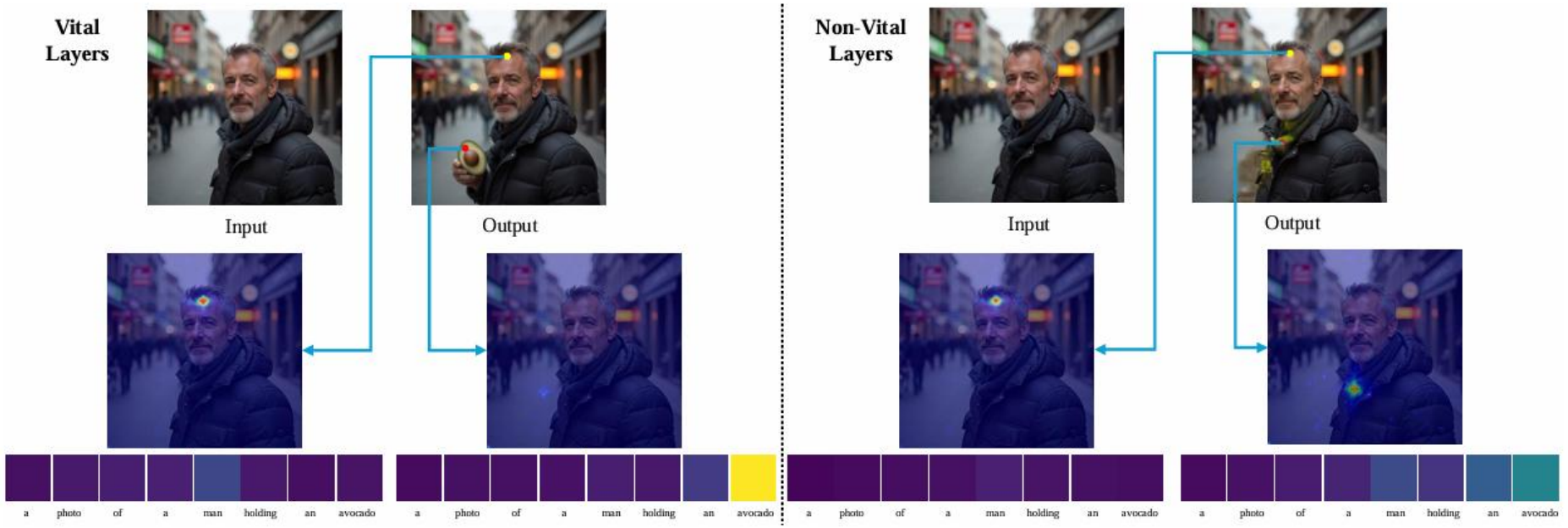


Methods - Stable Flow

- Selective Attention Injection
  - Vital Layers: Inject features (K, V) from Source
    - Preserve Structure & Identity
  - Non-Vital Layers: Generate new features from Edit Prompt
    - Function: Allow Semantic Changes

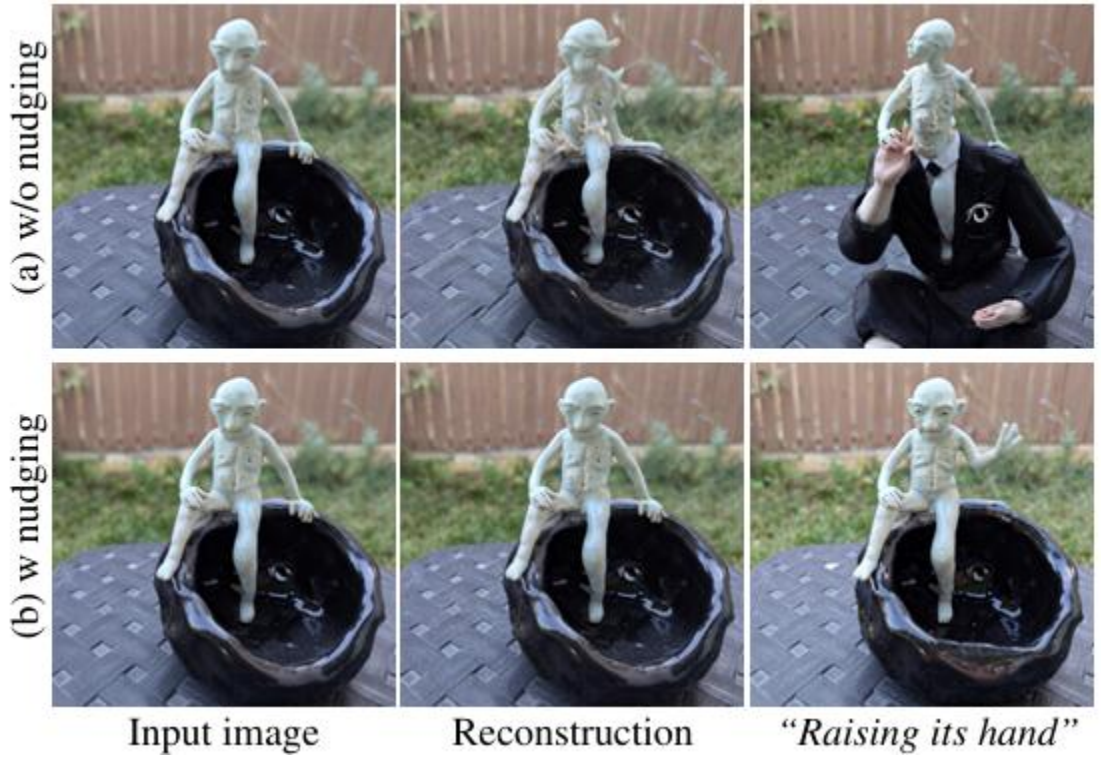
Methods - Stable Flow

- Vital Layers: Strong attention to Visual Tokens (Self-preservation)
- Non-Vital Layers: Stronger attention to Text Tokens (Prompt adherence)
- Injecting into Vital Layers enforces the source image's visual constraints, while leaving Non-Vital layers to attend to the new text instructions



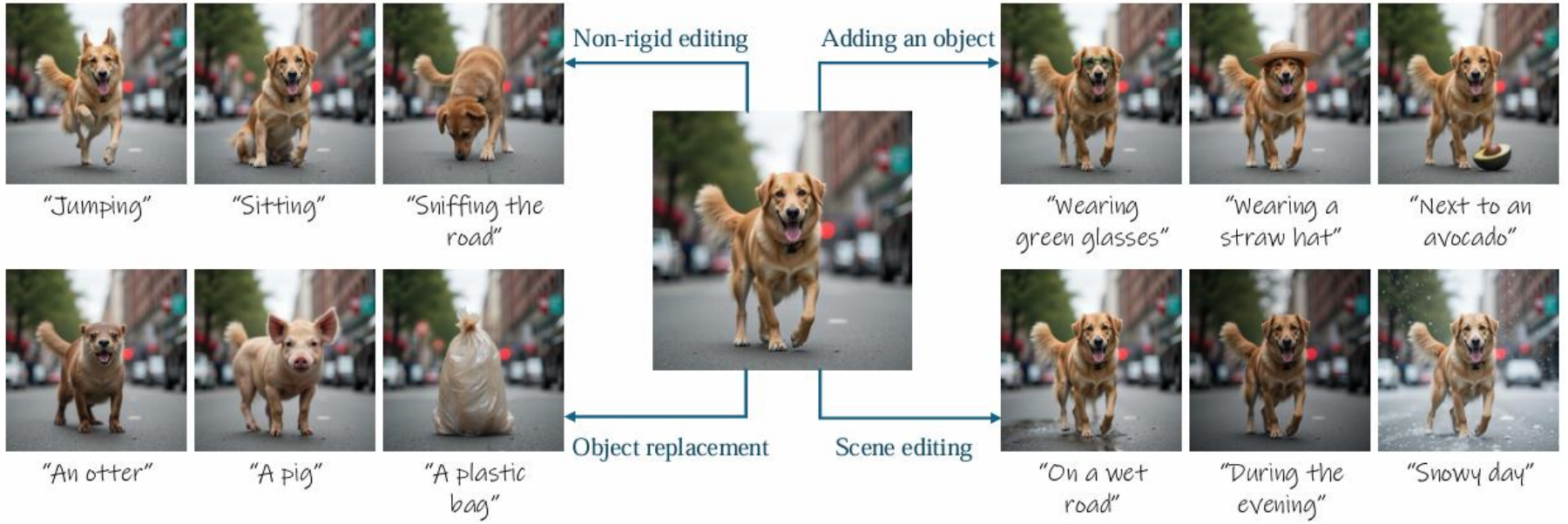
Methods - Stable Flow

- Problem: Euler Inversion creates artifacts $(u(z_t) \not\approx u(z_{t-1}))$
- Solution: Latent Nudging
  - Perturb initial latent z_0: $z_{new} = z_0 \times 1.15$
  - Push latent slightly Out-of-Distribution (OOD)



Methods - Stable Flow

- Capabilities
  - Non-Rigid Editing
  - Object Addition
  - Scene Editing
  - Style Transfer



Methods - Stable Flow

- Key Contribution: Automated detection of Vital Layers
- Strategy: Inject in Vital, Generate in Non-Vital
- Fix for Real Images: Latent Nudging
- Limitation: Doesn't differentiate types of editing

Outline

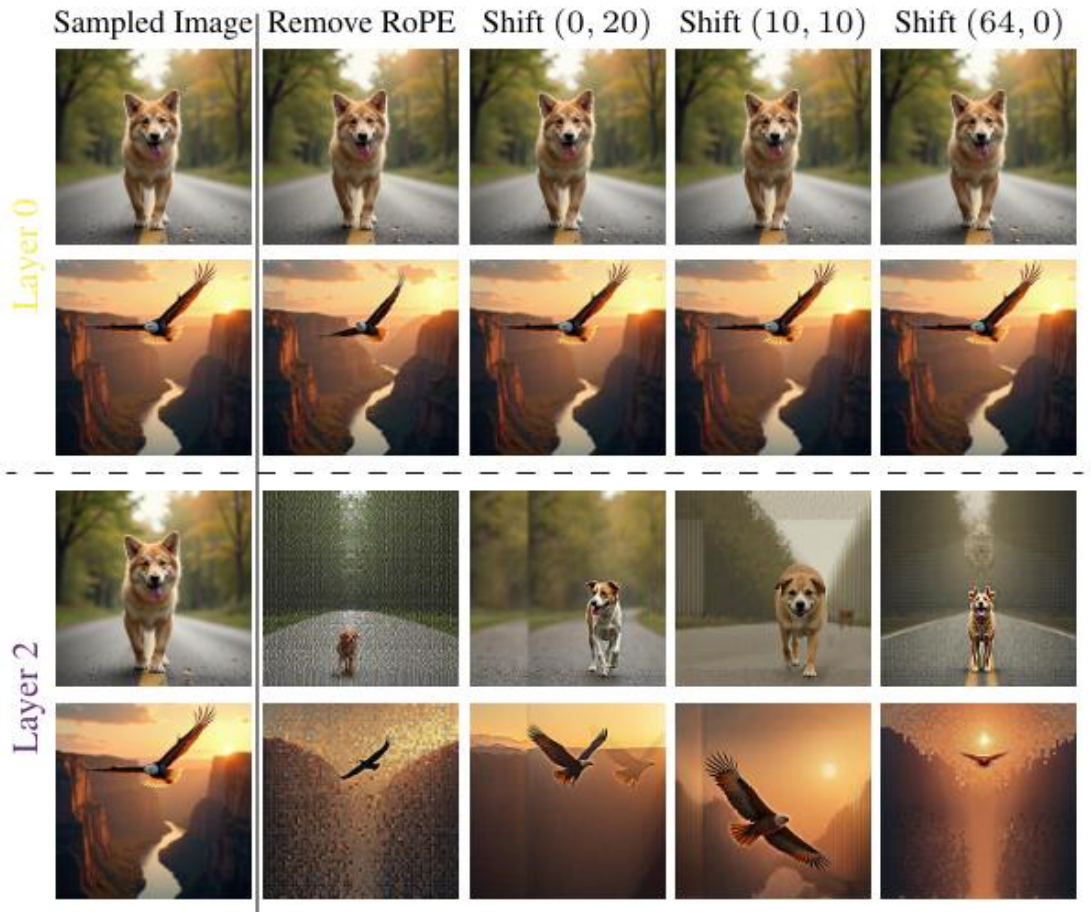
- Authors
- Background
- Stable Flow: Vital Layers Analysis
- FreeFlux: RoPE-Based Mechanism Analysis
- Experiments
- Conclusion

Methods - FreeFlux

- Attention Calculation: $Attn = Softmax(QK^T) \cdot V$
- With RoPE: $Attn = Softmax([Q_{txt}, RoPE(Q_{img})] \cdot [K_{txt}, RoPE(K_{img})]^T)$
- The Conflict: Does the attention mechanism rely on:
  - Semantic Similarity (Content matching)?
  - Positional Encoding (RoPE matching)?

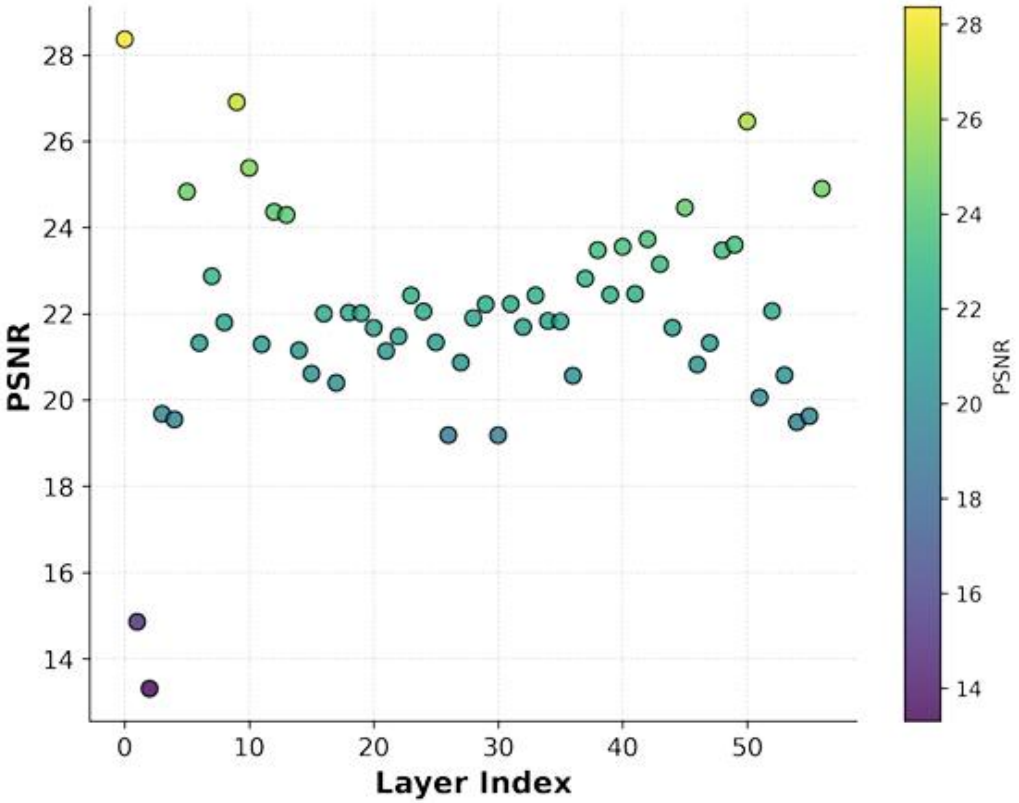
Methods - FreeFlux

- Method: Manipulate RoPE during inference to see what breaks
- Intervention:
  - Keep Q unchanged
  - Shift or Remove RoPE from K
- Logic:
  - If shifting RoPE ruins the image -> Layer relies on Position
  - If shifting RoPE has no effect -> Layer relies on Content



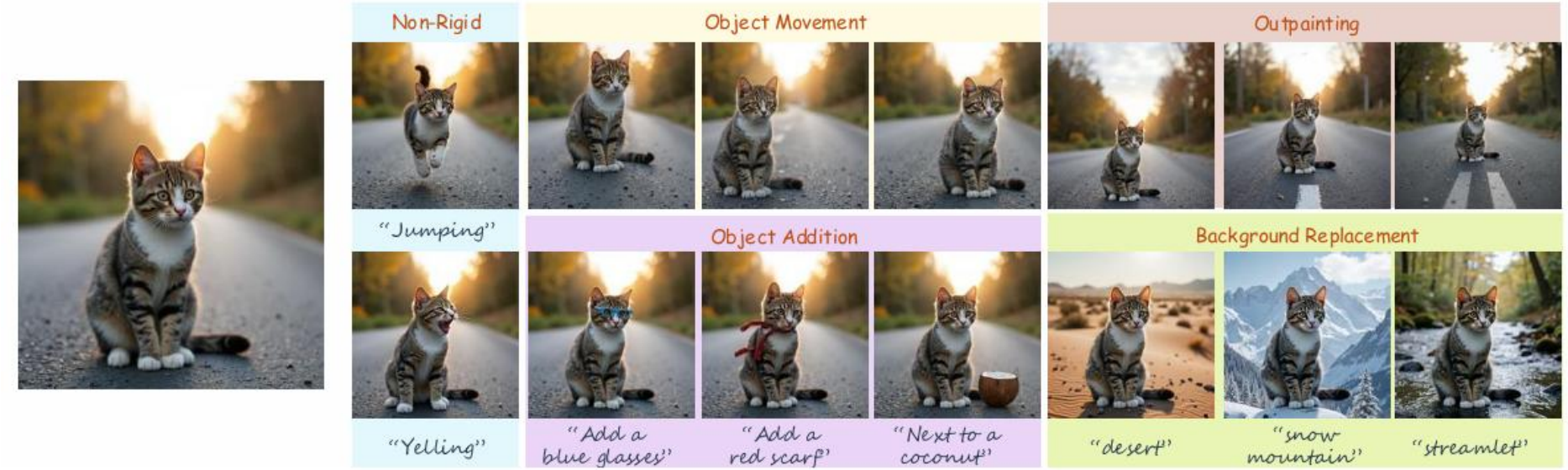
Methods - FreeFlux

- Position-Dependent Layers (P-Layers):
  - Image breaks if RoPE is touched
  - Role: Structure, Layout, Object Placement.
  - Indices: e.g., Layers 1, 2, 4, 26, 30...
- Content-Dependent Layers (C-Layers):
  - Image stable even without RoPE
  - Role: Texture, Semantics, Appearance.
  - Indices: e.g., Layers 0, 7, 8, 9, 10...



Methods - FreeFlux

- Idea: Different editing tasks require manipulating different information
- Categorization:
  - Position-Dependent Editing: Object Addition (Needs to place something new)
  - Content-Dependent Editing: Non-Rigid Editing (Change shape but keep texture)
  - Region-Preserved Editing: Background Replacement (Keep FG pixels exact)

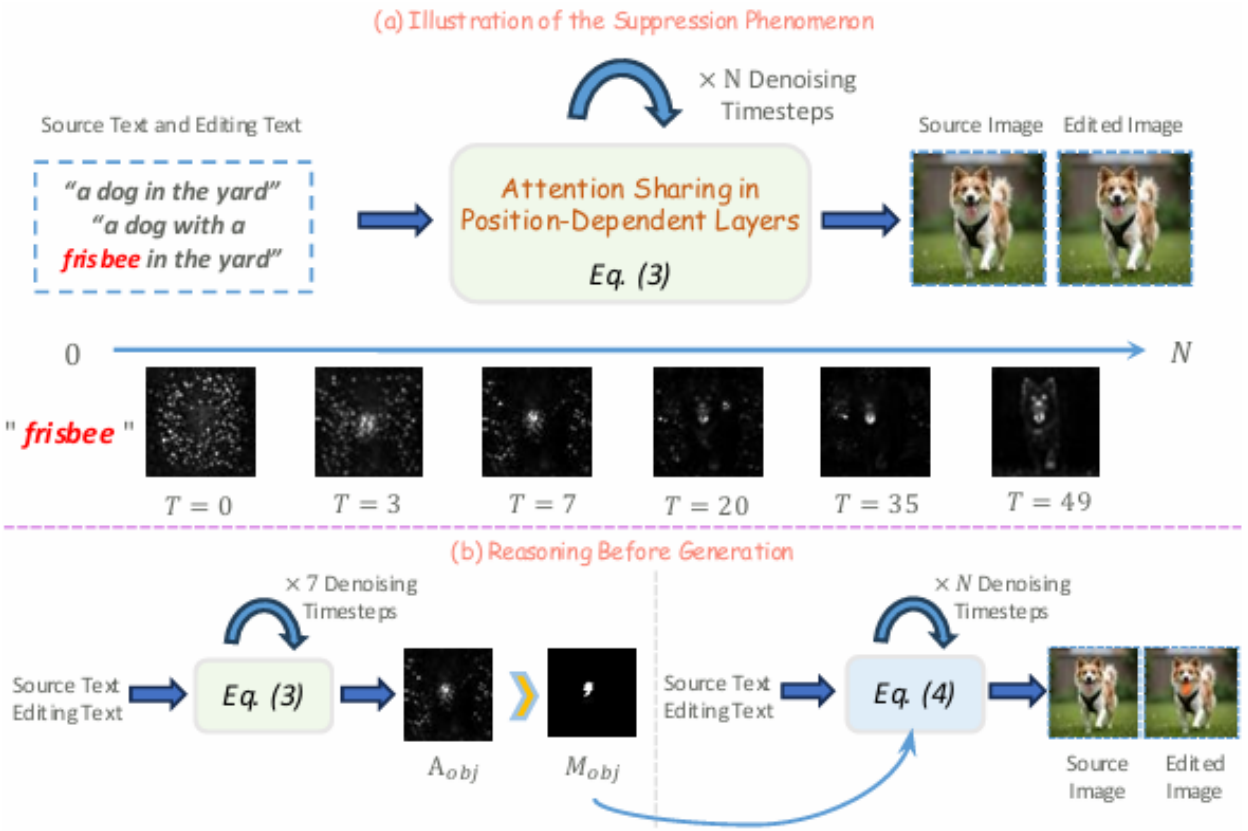


Methods - FreeFlux

- Task: Add a new object
- Strategy: Inject P-Layers
  - Keep the original layout (P-Layers), allow C-Layers to generate
- Challenge: Suppression
  - Source image has "nothing" at the target spot
  - Injecting source features suppresses the new object

Methods - FreeFlux

- Step 1 (Reasoning): Run a partial sampling step to let the model "plan" where the new object goes
- Step 2 (Generation): Restart sampling
  - Inside the new object mask: Do NOT inject source features
  - Outside the mask: Inject source features

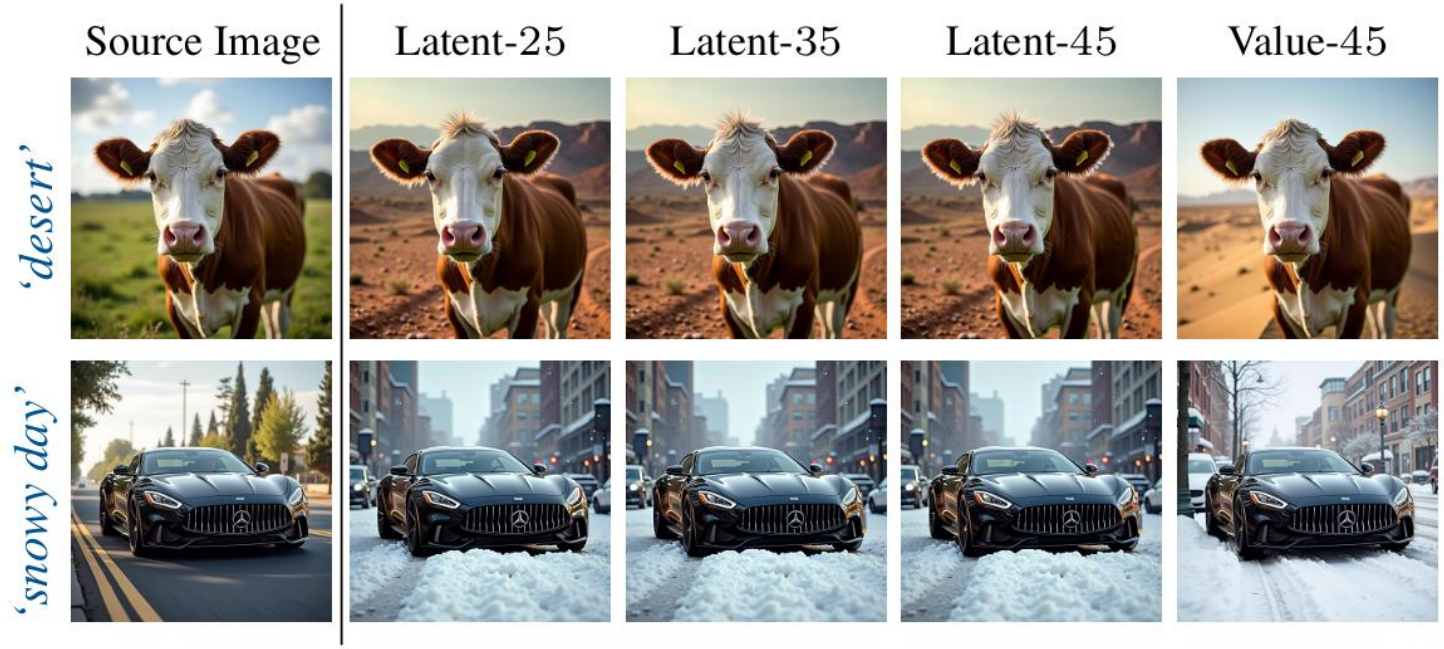


Methods - FreeFlux

- Task: Change pose
- Strategy: Inject C-Layers
  - Change Position
  - Keep Texture/Identity

Methods - FreeFlux

- Task: Change background, keep foreground
- Strategy: Value-Only Injection in ALL Layers
  - Use SAM-2 or Cross-Attention to mask foreground
  - Replace only V for foreground pixels
- Better edge blending than Latent Blending



Outline

- Authors
- Background
- Stable Flow: Vital Layers Analysis
- FreeFlux: RoPE-Based Mechanism Analysis
- Experiments
- Conclusion

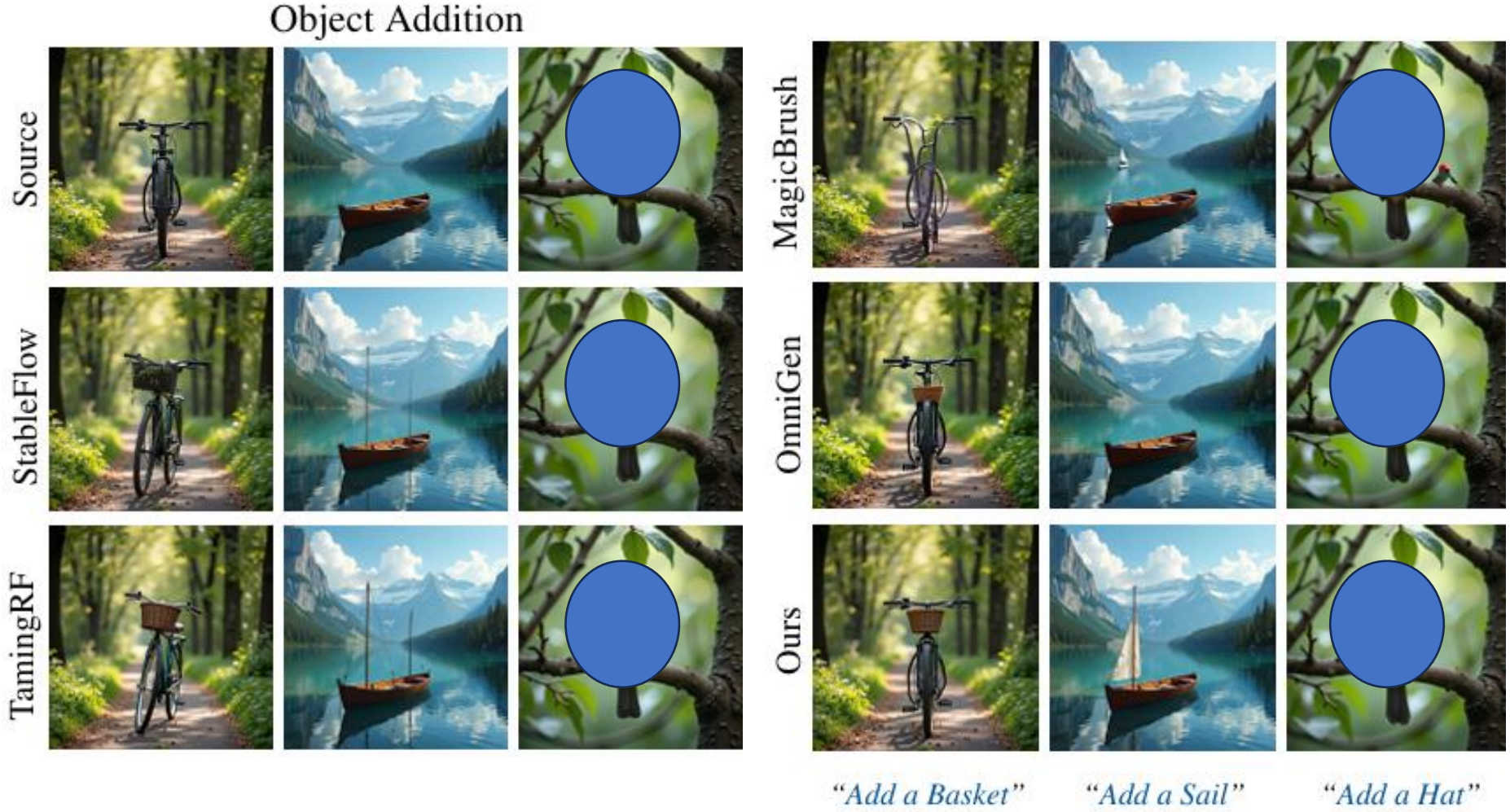
Experiments

Qualitative Results

- Comparison: Stable Flow vs. FreeFlux vs. InstructPix2Pix
- Observation:
  - Baselines often fail to change pose
  - FreeFlux (C-Layer injection): Successfully changes pose while keeping fur pattern

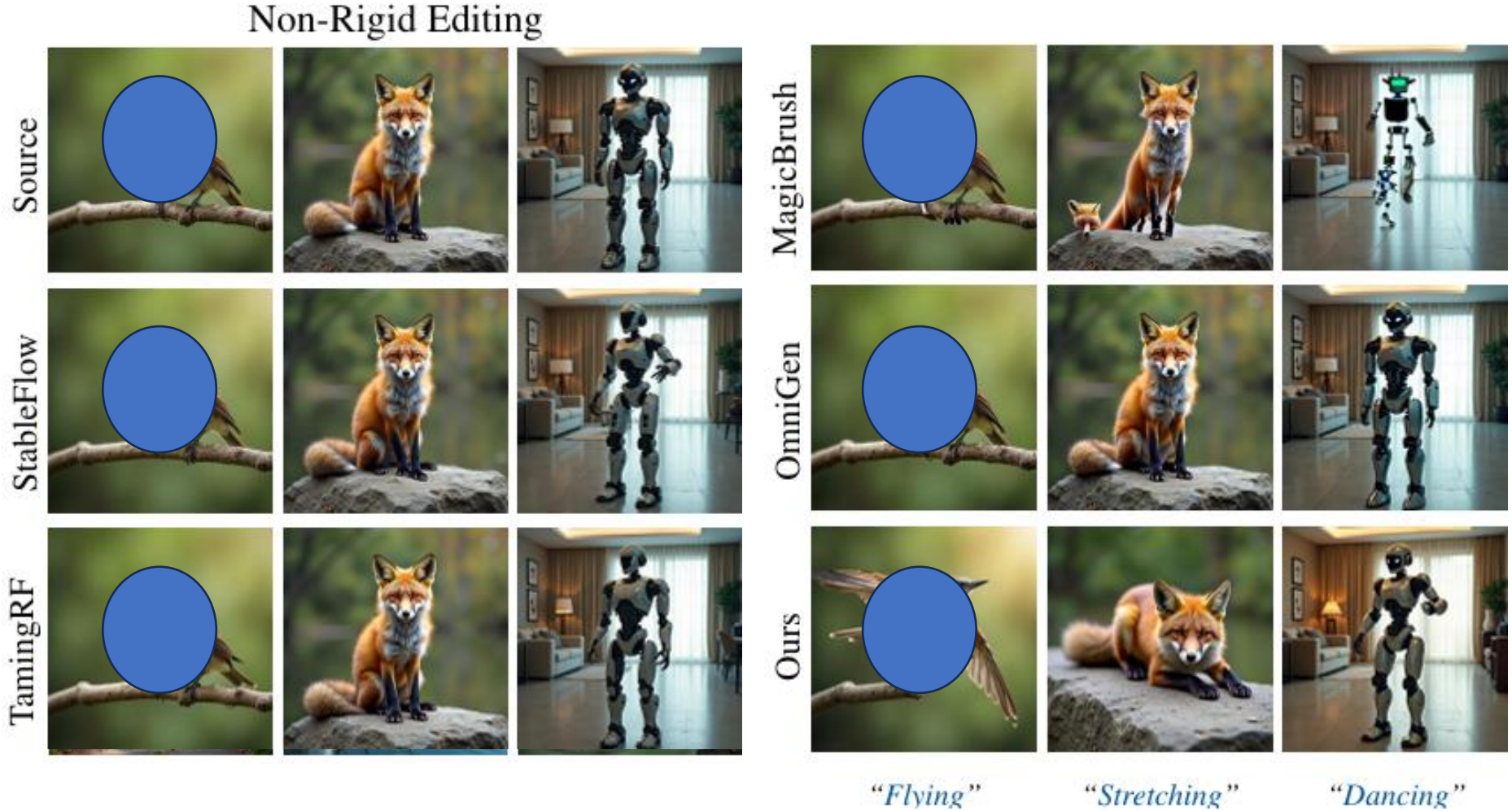
Experiments

Qualitative Results



Experiments

Qualitative Results



Experiments

Quantitative Results

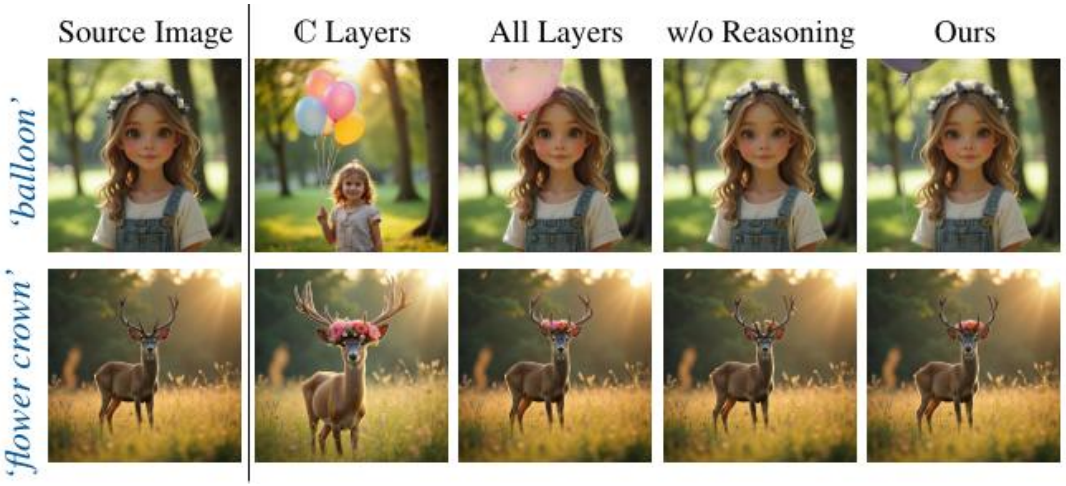
- Metrics: CLIP-Text (Prompt adherence), CLIP-Image (Structure preservation)
- Table: Compare Scores
  - Stable Flow achieves high balance
  - FreeFlux generally outperforms in CLIP-Dir (Directional change accuracy)

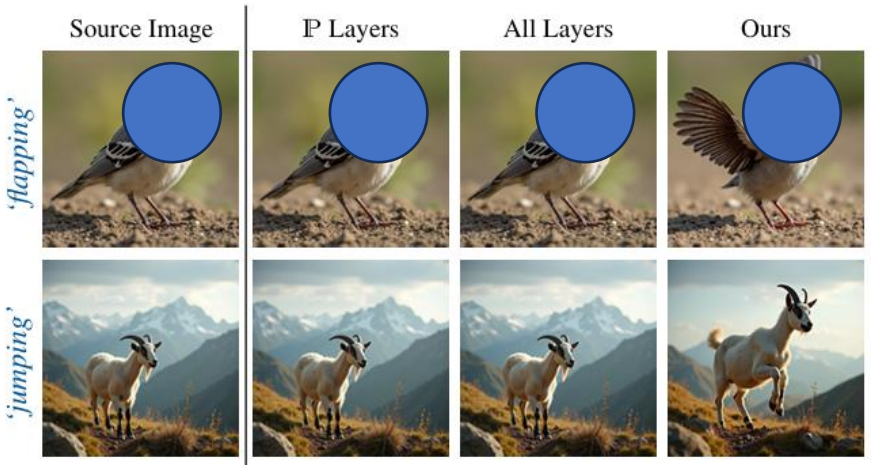
	Object Addition				Non-Rigid Editing				Background Replacement			
Methods	$\text{CLIP}_{img}\uparrow$	$\text{CLIP}_{txt}\uparrow$	$\text{CLIP}_{dir}\uparrow$	PR ↑	$\text{CLIP}_{img}\downarrow$	$\text{CLIP}_{txt}\uparrow$	$\text{CLIP}_{dir}\uparrow$	PR ↑	PSNR ↑	$\text{CLIP}_{txt}\uparrow$	$\text{CLIP}_{dir}\uparrow$	PR ↑
StableFlow	0.964	0.319	0.173	12.2%	0.969	0.307	0.124	11.1%	17.14	0.283	0.123	1.4%
TamingRF	0.958	0.320	0.175	31.9%	0.961	0.308	0.120	13.2%	16.32	0.284	0.134	1.9%
MagicBrush	0.944	0.319	0.161	1.6%	**0.933**	0.308	0.111	0.5%	14.87	0.308	0.261	13.0%
OmniGen	0.966	0.314	0.090	3.5%	0.974	0.303	0.047	1.1%	21.73	0.291	0.126	2.4%
Ours	**0.974**	**0.321**	**0.202**	**50.8%**	0.940	**0.315**	**0.153**	**74.1%**	**24.04**	**0.328**	**0.319**	**81.4%**

Experiments

Ablations

- Validation of P vs. C Layers
- Using P-Layers for Non-Rigid editing -> Failure (Image doesn't move).
- Using C-Layers for Object Addition -> Failure (Background changes).
- Conclusion: The mechanistic distinction is correct.





Outline

- Authors
- Background
- Stable Flow: Vital Layers Analysis
- FreeFlux: RoPE-Based Mechanism Analysis
- Experiments
- Conclusion

Conclusions

- Commonality:
  - DiT layers are heterogeneous (functionally different).
  - Parallel generation + Attention Injection is the way to go.
  - Flux requires special handling (Inversion, RoPE).
- Evolution:
  - Stable Flow: Empirical discovery ("Some layers are vital").
  - FreeFlux: Mechanistic explanation ("Layers differ by RoPE dependence").

Takeaways for DiT Editing

- Don't touch everything: Selective injection is key.
- Position vs. Semantics: Decoupled in Flux via RoPE.
- Pre-computation helps: Reasoning steps (Mask extraction) improve complex edits.

Thank you for Listening!