FlowEdit: Inversion-Free Text-Based Editing Using Pre-Trained Flow Models

V. Kulikov, M. Kleiner, I. Huberman-Spiegelglas, T. Michaeli

Presented by: **Yifan Liu, Mohammad Moshtaghifar, Siddharth Rout**

April 1, 2026

Outline

1. Introduction and Motivation
2. Review of Flow Models
3. Image Editing Using ODE Inversion
4. FlowEdit
5. Experiments
6. Conclusion

Outline

1. Introduction and Motivation
2. Review of Flow Models
3. Image Editing Using ODE Inversion
4. FlowEdit
5. Experiments
6. Conclusion

Problem Setup: Text-Based Editing of Real Images

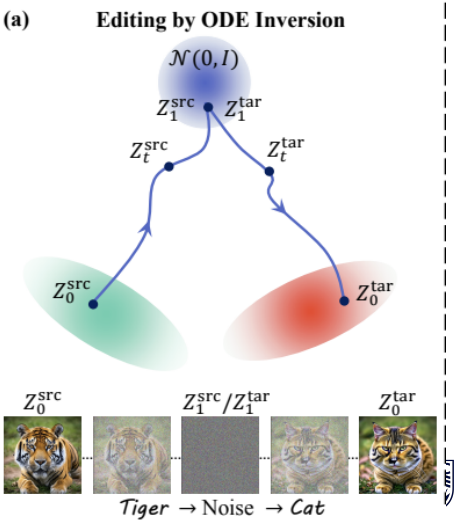
- This paper studies **text-based editing of real images** using a pre-trained text-to-image **flow model**.
- The input consists of:
  - a source image X_{src}
  - a source prompt c_{src}
  - a target prompt c_{tar}
- The goal is to produce an edited image that satisfies two requirements:
  - **Text adherence**: the output should follow the target prompt.
  - **Structure preservation**: the output should preserve the layout, identity, background, and local details of the source image whenever possible.
- In other words, a good editor should change **only what the prompt asks for** while keeping everything else stable.

Why Is This Problem Important?

- Real-image editing is more constrained than image generation.
- Users do not want a completely new image with similar semantics; they want a **controlled modification** of the original image.
- Typical editing tasks include:
  - changing an object or attribute
  - replacing scene text
  - modifying style while preserving content
- The core challenge is balancing two competing goals:
  - strong editability
  - high fidelity to the source image
- Failure is usually obvious: the edit may satisfy the text prompt but unintentionally alter the background, pose, geometry, or identity.

Existing Paradigm: Editing by Inversion

- A common zero-shot pipeline is **editing by inversion**.
- Step 1: invert the source image into a noise representation.
- Step 2: use the same noise and sample back with a new target prompt.
- In flow models, this corresponds to:
  - a forward ODE: image $\rightarrow$ noise
  - a reverse ODE: noise $\rightarrow$ edited image
- The intuition is simple: keep the latent/noise fixed, and change only the text condition.



Why Inversion Is Not Enough

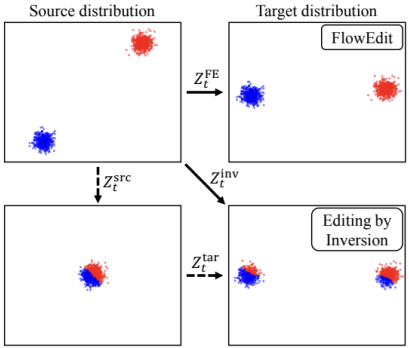
- The paper argues that **editing-by-inversion often fails to preserve source fidelity**.
- Improving inversion accuracy alone does **not fully solve** the problem.
- Even with **exact inversion** on synthetic data, the edited image may still deviate too much from the source structure.
- To address this, many prior methods inject internal signals such as:
  - attention maps
  - intermediate feature representations
- These methods may improve fidelity, but they are often:
  - architecture-specific
  - sampler-specific
  - difficult to transfer across models
- Therefore, the issue is not only inaccurate inversion; it is also the **editing paradigm itself**.

Motivation: A Shorter Path from Source to Target

- Inversion-based editing follows an indirect route:

$$\text{source image} \rightarrow \mathcal{N}(0, I) \rightarrow \text{target image}$$

- This route may induce a **high transport cost**, meaning the source image is modified more than necessary.
- A better editing method should:
  - move the source image into the target distribution
  - while changing it **as little as possible**
- The key motivation of the paper is:
  - avoid passing through Gaussian noise
  - instead build a **direct path** between the source and target distributions



FlowEdit aims for lower transport cost

FlowEdit: High-Level Idea and Contributions

- **FlowEdit** is an editing method for pre-trained text-to-image flow models that is:
  - **inversion-free**
  - **optimization-free**
  - **model-agnostic**
- The main conceptual insight is:
  - reinterpret inversion as a direct path between source and target distributions
  - then construct a better direct path with **lower transport cost**
- Empirically, the paper shows that FlowEdit achieves:
  - stronger structure preservation
  - strong text adherence
  - state-of-the-art performance on diverse editing tasks
- The method is validated on both **Stable Diffusion 3** and **FLUX**.
- **Next question:** how is this direct ODE path constructed?

Take-Home Message

Core Motivation

The key idea of the paper is to replace an indirect inversion-based editing path with a shorter and more structure-preserving direct path.

- This explains why FlowEdit can better preserve the structure of the source image.
- It also motivates the technical development in the next section.

Outline

1. Introduction and Motivation
2. Review of Flow Models
3. Image Editing Using ODE Inversion
4. FlowEdit
5. Experiments
6. Conclusion

Generative Flow Models – Setup

- Goal: transport between two distributions X_0 (data) and X_1 (noise) via an ODE:

$$dZ_t = V(Z_t, t)\, dt, \quad t \in [0, 1] \tag{1}$$

- V is a learned, time-dependent **velocity field** (neural network).
- Boundary condition: $Z_1 = X_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \Rightarrow Z_0 = X_0 \sim p_{\text{data}}$.
- Solving the ODE backward ($t : 1 \to 0$) maps noise to data.

Generative Flow Models – Setup

- Goal: transport between two distributions X_0 (data) and X_1 (noise) via an ODE:

$$dZ_t = V(Z_t, t)\, dt, \quad t \in [0, 1] \tag{1}$$

- V is a learned, time-dependent **velocity field** (neural network).
- Boundary condition: $Z_1 = X_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \Rightarrow Z_0 = X_0 \sim p_{\text{data}}$.
- Solving the ODE backward ($t : 1 \to 0$) maps noise to data.

Sampling:

1. Draw $Z_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
2. Solve the ODE backward in time.
3. Obtain $Z_0 \sim p_{\text{data}}$.

Rectified Flow Models

Rectified Flows [Liu et al., 2022] are a particular family of flow models where the marginal at time t is a **linear interpolation**:

$$Z_t \sim (1-t)\, X_0 + t\, X_1 \tag{2}$$

Key property: sampling paths are relatively **straight**, so a small number of ODE discretization steps suffices.

Text-Conditioned Generation

- The velocity field is conditioned on a text prompt C:

$$V(Z_t, t, C)$$

- Trained on (C, X_0) pairs $\Rightarrow$ can sample from $X_0 \mid C$.
- Examples: Stable Diffusion 3 [Esser et al., 2024], FLUX.

Outline

1. Introduction and Motivation
2. Review of Flow Models
3. Image Editing Using ODE Inversion
4. FlowEdit
5. Experiments
6. Conclusion

Editing by Inversion – The Standard Approach

Given:

- Source image X^{src} with source prompt c^{src} and target prompt c^{tar}.

Notation:

$$V^{\text{src}}(Z_t, t) \triangleq V(Z_t, t, c^{\text{src}}), \qquad V^{\text{tar}}(Z_t, t) \triangleq V(Z_t, t, c^{\text{tar}})$$

Editing by Inversion – The Standard Approach (contd.)

Two-step procedure:

1. **Forward (inversion):** Solve

$$dZ_t^{\text{src}} = V^{\text{src}}(Z_t^{\text{src}}, t)\, dt \quad \text{forward}\,(t: 0 \to 1)$$

starting from $Z_0^{\text{src}} = X^{\text{src}}$.

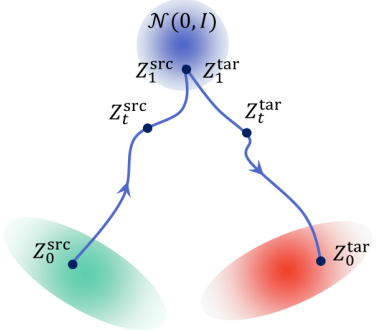
$\Rightarrow$ Obtain noise map Z_1^{src}.

2. **Reverse (sampling):** Solve

$$dZ_t^{\text{tar}} = V^{\text{tar}}(Z_t^{\text{tar}}, t)\, dt \quad \text{backward}\,(t: 1 \to 0)$$

starting from $Z_1^{\text{tar}} = Z_1^{\text{src}}$.

$\Rightarrow$ Obtain edited image Z_0^{tar}.



Editing by Inversion - Limitations

$$\textbf{Source} \xrightarrow[\text{forward ODE w/ } c^{\text{src}}]{} \textbf{Noise} \xrightarrow[\text{reverse ODE w/ } c^{\text{tar}}]{} \textbf{Target}$$

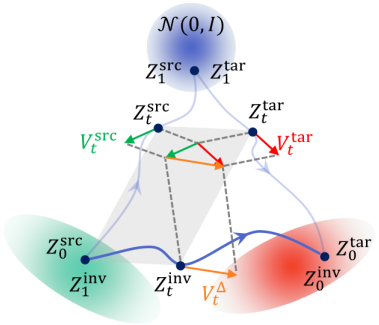
- Poor structure preservation.
- Even with exact inversion (known noise), results are often unsatisfactory.
- Feature injection methods are architecture-specific and not transferable.

Reinterpretation of Editing by Inversion

Key insight: Editing by inversion can be rewritten as a **direct path** between source and target distributions, without explicitly passing through noise.

Define:

$$\boxed{Z_t^{\text{inv}} = Z_0^{\text{src}} + Z_t^{\text{tar}} - Z_t^{\text{src}}} \tag{5}$$



Reinterpretation of Editing by Inversion

Key insight: Editing by inversion can be rewritten as a **direct path** between source and target distributions, without explicitly passing through noise.

Define:

$$\boxed{Z_t^{\text{inv}} = Z_0^{\text{src}} + Z_t^{\text{tar}} - Z_t^{\text{src}}} \tag{5}$$

Verify the boundary conditions:

- At $t = 1$: $Z_1^{\text{inv}} = Z_0^{\text{src}} + \underbrace{Z_1^{\text{tar}} - Z_1^{\text{src}}}_{=0} = X^{\text{src}}$ (source image).
- At $t = 0$: $Z_0^{\text{inv}} = Z_0^{\text{src}} + Z_0^{\text{tar}} - Z_0^{\text{src}} = Z_0^{\text{tar}}$ (edited image).

$\Rightarrow$ Going from $t = 1$ to $t = 0$, Z_t^{inv} transitions from the source image to the edited image **directly**.

From Direct Path to ODE

- Differentiating $Z_t^{\text{inv}} = Z_0^{\text{src}} + Z_t^{\text{tar}} - Z_t^{\text{src}}$:

$$dZ_t^{\text{inv}} = V_t^{\Delta}(Z_t^{\text{src}}, Z_t^{\text{tar}})\, dt \tag{6}$$

where

$$V_t^{\Delta}(Z_t^{\text{src}}, Z_t^{\text{tar}}) = V^{\text{tar}}(Z_t^{\text{tar}}, t) - V^{\text{src}}(Z_t^{\text{src}}, t)$$

- Removing the explicit dependence on Z_t^{tar}:

$$\boxed{dZ_t^{\text{inv}} = V_t^{\Delta}\big(Z_t^{\text{src}},\ Z_t^{\text{inv}} + Z_t^{\text{src}} - Z_0^{\text{src}}\big)\, dt}$$

with boundary condition $Z_1^{\text{inv}} = Z_0^{\text{src}}$ at $t = 1$.

The editing direction $= V^{\text{tar}}(\cdot) - V^{\text{src}}(\cdot)$ (target velocity minus source velocity).

Properties of the Direct Path

Why is this path special?

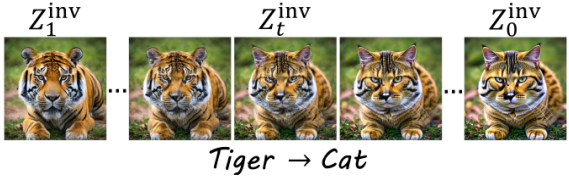
- **Noise-free:** The noisy images Z_t^{tar} and Z_t^{src} contain roughly the same noise. Their velocities V^{tar} and V^{src} remove the same noise component, so:

$$V_t^{\Delta}(Z_t^{\text{src}}, Z_t^{\text{tar}}) = V^{\text{tar}}(Z_t^{\text{tar}}, t) - V^{\text{src}}(Z_t^{\text{src}}, t)$$

  captures only the difference between **clean image predictions**.

- **Coarse-to-fine evolution:**
  - $t \approx 1$ (high noise): V_t^{Δ} captures only **coarse structure** differences.
  - $t \to 0$ (low noise): high-frequency details are unveiled.

  $\Rightarrow$ Autoregressive coarse-to-fine transition from source to target.



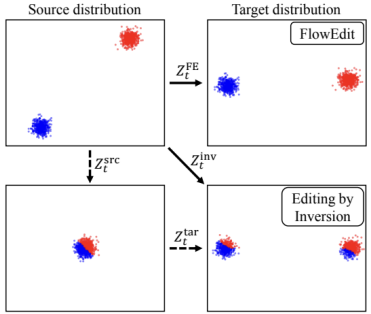
Outline

1. Introduction and Motivation
2. Review of Flow Models
3. Image Editing Using ODE Inversion
4. FlowEdit
5. Experiments
6. Conclusion

Inversion to Flowedit

Problem with Inversion

- Editing-by-inversion induces **suboptimal pairings** between source and target samples
- Leads to:
  - Large, unnecessary changes
  - Poor structure preservation
- Root cause: pairings are dictated by mapping to **initial noise**



Goal

- Construct mappings that minimize distance between source and target samples
- Preserve structure by mapping to **nearest valid targets**

Key Idea (FlowEdit)

- Replace single inversion path with **multiple random pairings**
- Average their velocity fields

FlowEdit: Improving Sample Pairing Beyond Inversion

Method

- Define alternative forward process:

$$\hat{Z}_t^{\text{src}} = (1-t)Z_0^{\text{src}} + tN_t, \quad N_t \sim \mathcal{N}(0,1)$$

- Solve ODE with **expected velocity**:

$$dZ_t^{\text{FE}} = \mathbb{E}[V_t^{\Delta}(\cdot) \mid Z_0^{\text{src}}]\, dt$$

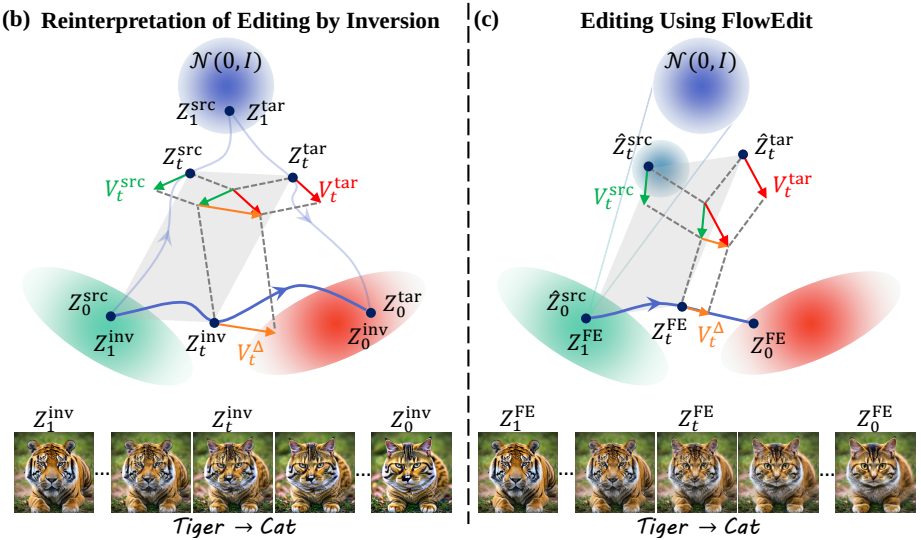
Outcome

- Better (near-nearest) sample pairings
- Lower transport cost ($< 2\times$ vs inversion)
- Improved structure preservation in image editing

Note

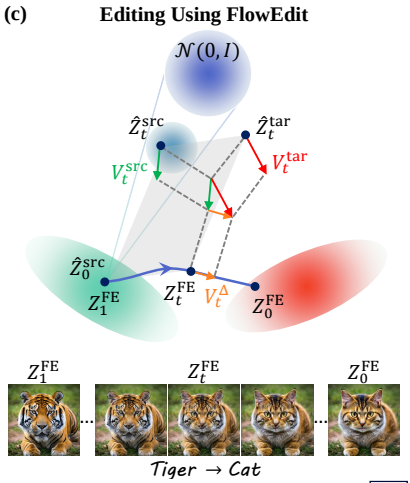
- Heuristic method (not exact optimal transport), but strong empirical results

Visualizing FlowEdit



FlowEdit: Simplified Algorithm

Input: real image X^{src}, $\{t_i\}_{i=0}^{T}$, $n_{\max}$, n_{avg}
Output: edited image X^{tar}
Init: $Z^{\text{FE}}_{t_{\max}} = X^{\text{src}}_0$
for $i = n_{\max}$ **to** 1 **do**
  for $j = n_{\text{avg}}$ **to** 1 **do**
    $N_{t_{ij}} \sim \mathcal{N}(0, 1)$
    $Z^{\text{src}}_{t_{ij}} \leftarrow (1 - t_i)X^{\text{src}} + t_i N_{t_{ij}}$
    $Z^{\text{tar}}_{t_{ij}} \leftarrow Z^{\text{FE}}_{t_i} + Z^{\text{src}}_{t_{ij}} - X^{\text{src}}$
    $V^{\Delta}_{t_{ij}} \leftarrow V^{\text{tar}}(Z^{\text{tar}}_{t_{ij}}, t_i) - V^{\text{src}}(Z^{\text{src}}_{t_{ij}}, t_i)$
  end for
  $V^{\Delta}_{t_i} \leftarrow \Sigma_{j=1}^{n_{avg}} V^{\Delta}_{t_{ij}} / n_{avg}$
  $Z^{\text{FE}}_{t_{i-1}} \leftarrow Z^{\text{FE}}_{t_i} + (t_{i-1} - t_i)V^{\Delta}_{t_i}$
end for
Return: $Z^{\text{FE}}_0 = X^{\text{tar}}_0$



Outline

1. Introduction and Motivation
2. Review of Flow Models
3. Image Editing Using ODE Inversion
4. FlowEdit
5. Experiments
6. Conclusion

Comparing SD3[Esser et al., 2024] and FLUX[Labs et al., 2025] Pre-trained on DIV2K [Agustsson and Timofte, 2017]

Comparison Metrics with SOTA Methods

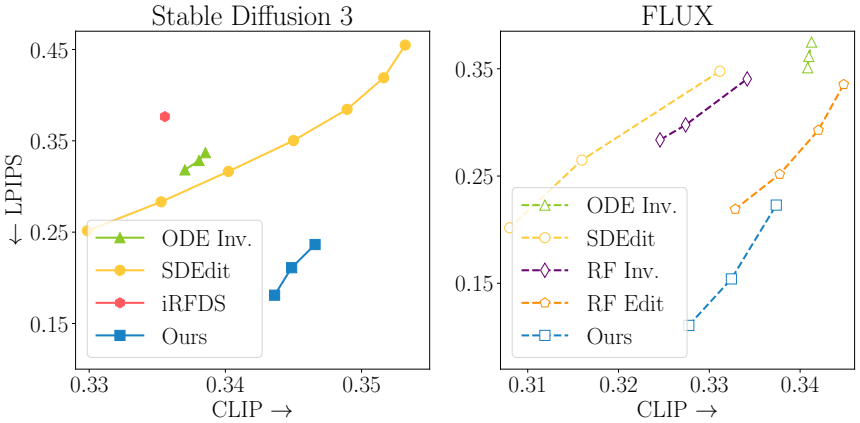
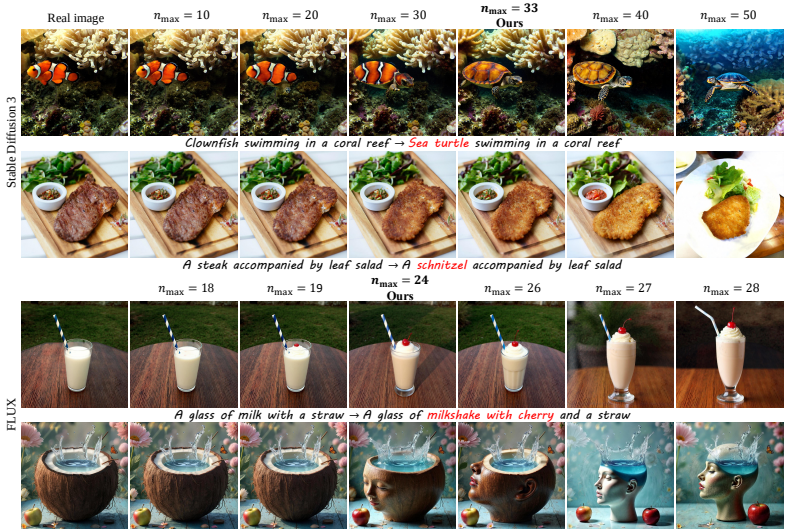


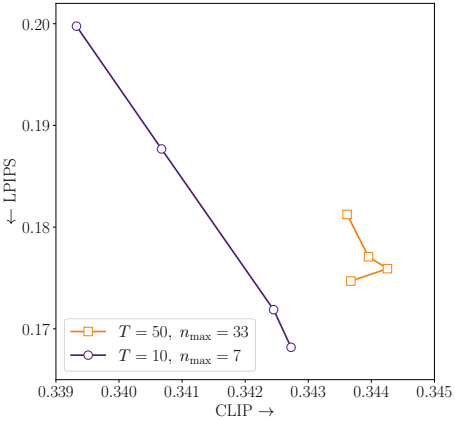
Figure: FlowEdit achieves a favorable balance between text adherence (CLIP) and structure preservation (LPIPS) compared to other methods.

Dependence on Hyperparameters: n_{max}



Dependence on Hyperparameters: n_{avg}

Figure: From top to bottom, the markers correspond to n_{avg} of 1, 3, 5, 10.



Outline

1. Introduction and Motivation
2. Review of Flow Models
3. Image Editing Using ODE Inversion
4. FlowEdit
5. Experiments
6. Conclusion

Conclusion

- **Strong qualitative performance**
- **Outperforms prior methods**
- **Style editing**
  - Supports diverse styles (anime, watercolor, etc.)
  - Trade-off: slight loss of structure sometimes
- **Limitation**
  - Less effective for large/global changes (pose, background)
- **Takeaway**
  - Strong balance between edit accuracy and image fidelity
  - Strong dependence on the introduced hyperparameters (specifically, n_{avg})

Open Questions

- **Adaptive control of the editability–fidelity trade-off.** The method's quality is sensitive to hyperparameters such as n_{avg} and t_{max}, which must currently be set manually. An open direction could be to check whether the right trade-off can be predicted automatically. For example, we could use the semantic distance between source and target prompts.
- Fixing n_{max} feels counterintuitive from a theoretical perspective, so there can be a study identifying why it fails at larger n_{max}.

References

Agustsson, E. and Timofte, R. (2017).
Ntire 2017 challenge on single image super-resolution: Dataset and study.
In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Muller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Lacey, K., Goodwin, A., Marek, Y., and Rombach, R. (2024).
Scaling rectified flow transformers for high-resolution image synthesis.
In *International Conference on Machine Learning*.

Labs, B. F., Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., Kulal, S., Lacey, K., Levi, Y., Li, C., Lorenz, D., Müller, J., Podell, D., Rombach, R., Saini, H., Sauer, A., and Smith, L. (2025).
Flux.1 kontext: Flow matching for in-context image generation and editing in latent space.

Liu, X., Gong, C., and Liu, Q. (2022).
Flow straight and fast: Learning to generate and transfer data with rectified flow.

Thank You!

Questions?