Idempotent Generative Network

Assaf Shocher Amil Dravid Yossi Gandelsman

Inbar Mosseri Michael Rubinstein Alexei A. Efros

STRUCT Group Seminar
Presenter: Haowei Kuang
2023.11.19

ArXiv 2311(Submitted to ICLR 2024)

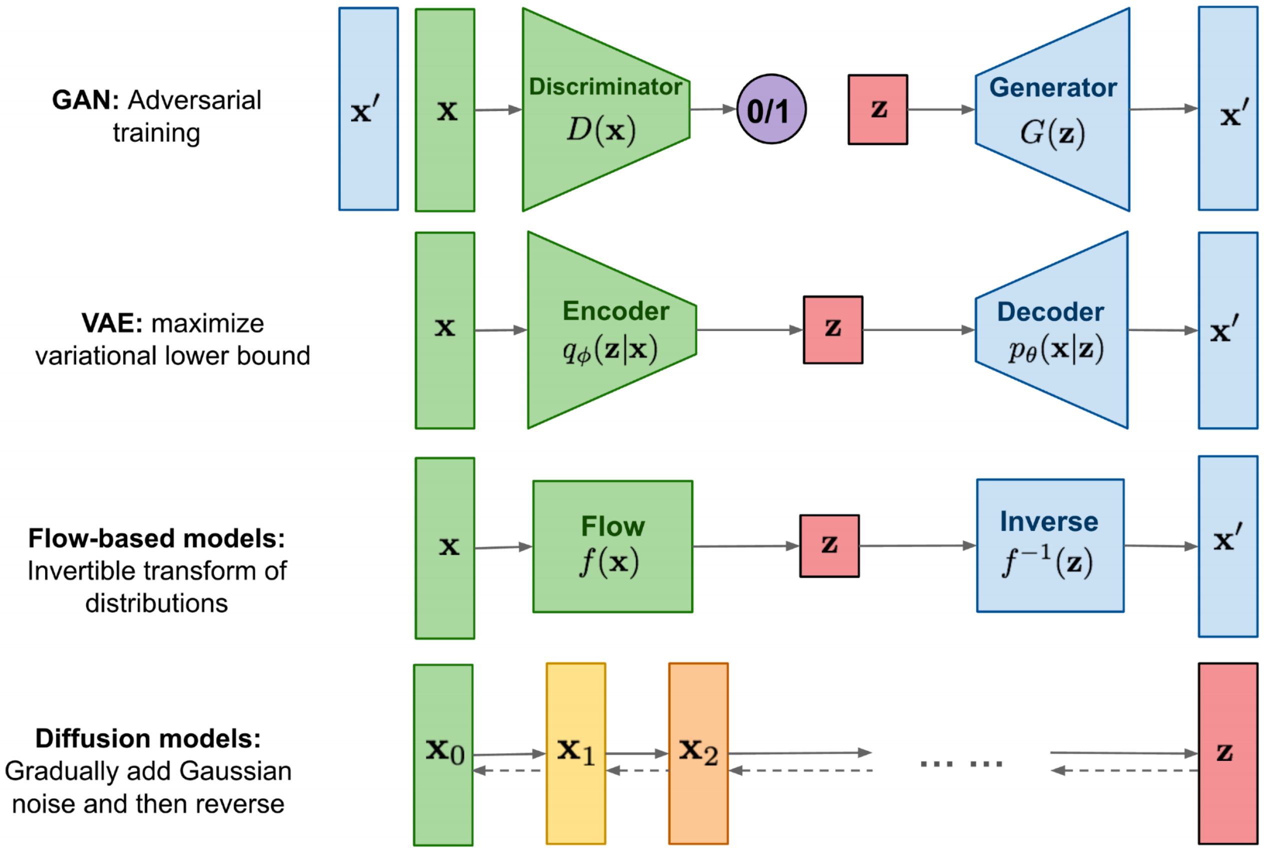
OUTLINE

- Background
- Method
- Experiments
- Conclusion

BACKGROUND: Generative Models

What is generative models?

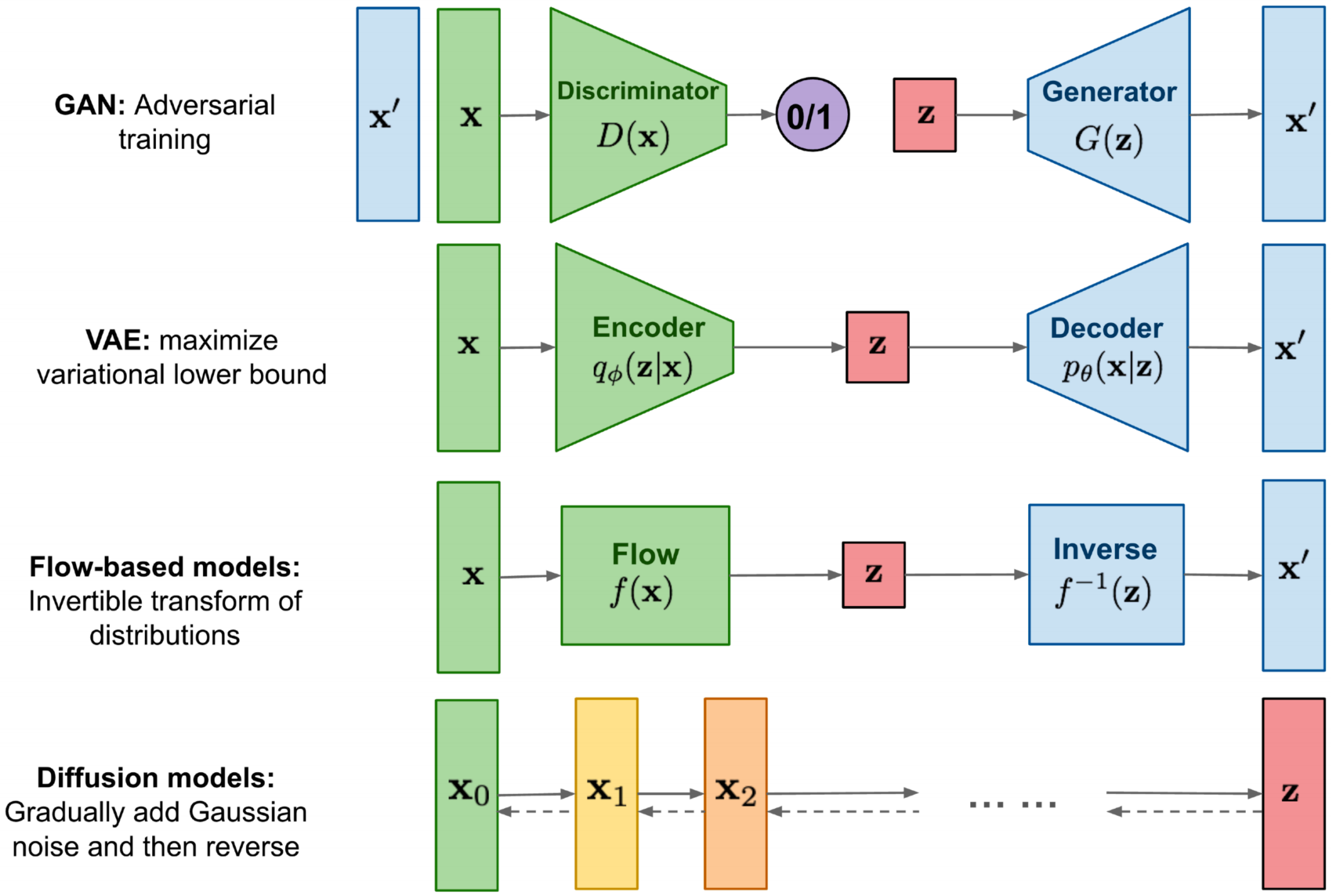
What are in generative model family?



BACKGROUND: Generative Models

What's the ideal generative models?

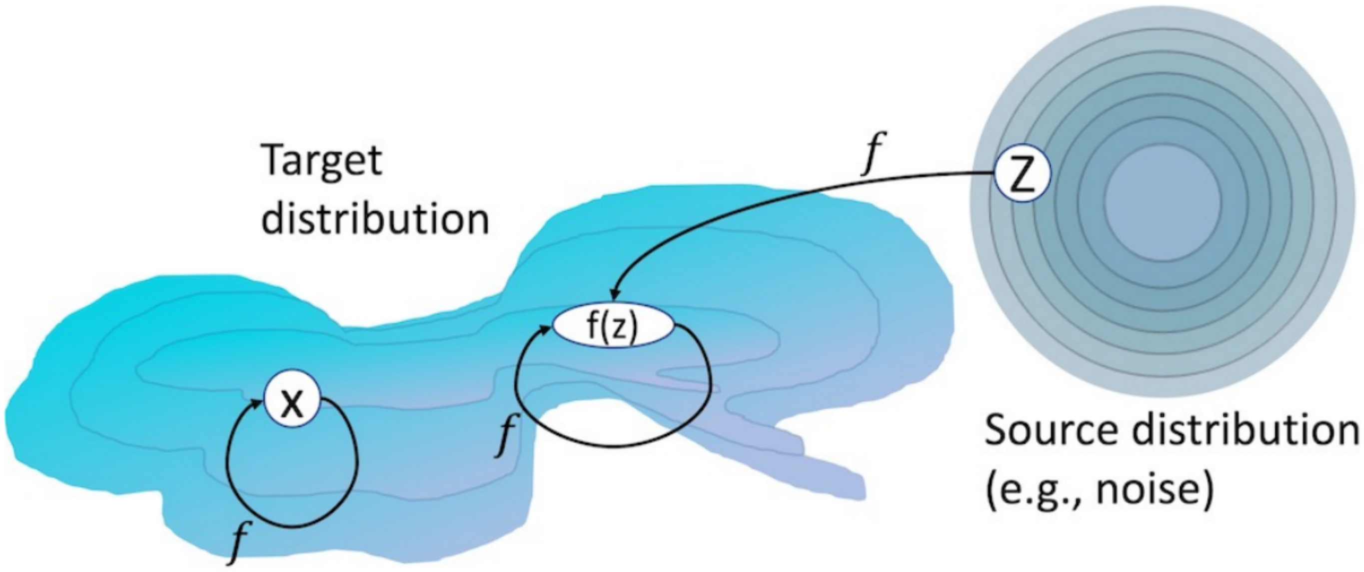
- One step projector
- Global projector



BACKGROUND: Generative Models

A novel generative model —— Idempotent Generative Network

- The first step towards a “global projector”



BACKGROUND: Idempotent

Applied sequentially multiple times without changing the result beyond the initial application:

$$f(f(z)) = f(z)$$

Examples: $||z|| = |z|$ Orthogonal Projection $A^2 = A$

GEORGE: *You're gonna "overdry" it.*
JERRY: *You, you can't "overdry."*
GEORGE: *Why not?*
JERRY: *Same as you can't "overwet." You see, once something is wet, it's wet. Same thing with dead: like once you die you're dead, right? Let's say you drop dead and I shoot you: you're not gonna die again, you're already dead. You can't "overdie," you can't "overdry."*

— *"Seinfeld", Season 1, Episode 1, NBC 1989*

OUTLINE

- Background
- Method
- Experiments
- Conclusion

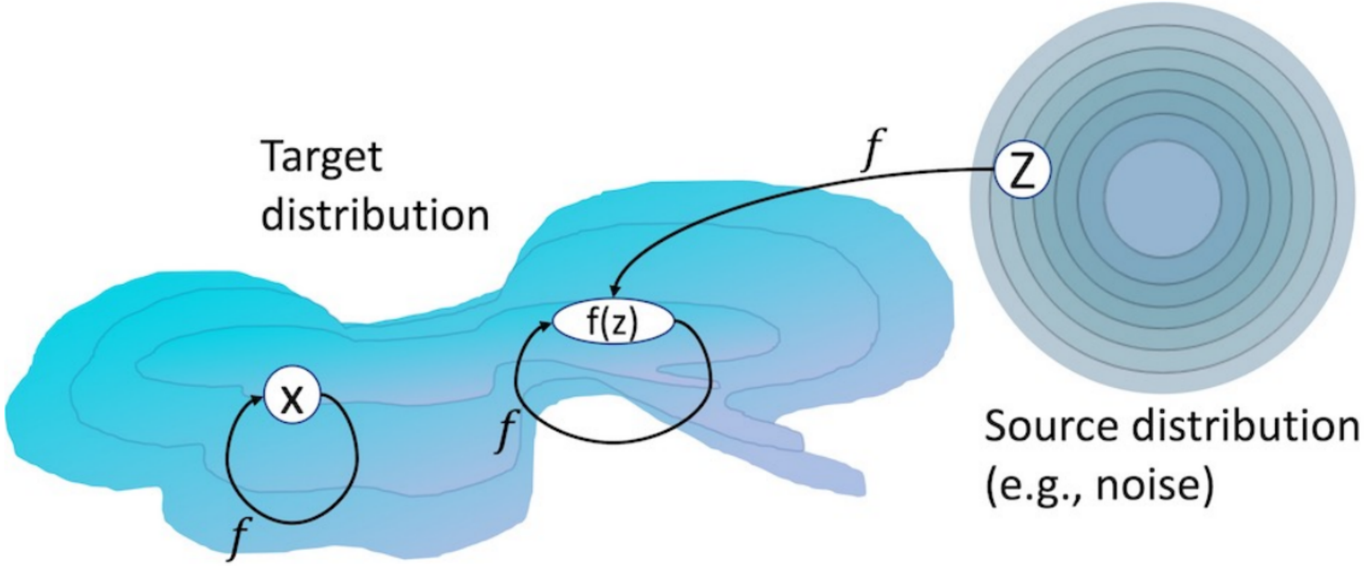
METHOD: Overview

Generate samples from a target distribution $\mathcal{P}_x$ given input from a source distribution $\mathcal{P}_z$

Basic idea:

Learning a model f satisfy:

$$f(x) = x$$
$$f(f(z)) = f(z)$$



METHOD: Optimization Objectives

Reconstruction objective

Each sample $x \sim \mathcal{P}_x$ is mapped to itself: $f(x) = x$

Define the drift measure of some instance y as: $\delta_\theta(y) = D\big(y, f_\theta(y)\big)$

$$\mathcal{S} = \{y : f(y) = y\} = \{y : \delta(y) = 0\}$$

Idempotent objective

Similarly, we hope $f(z) \in \mathcal{S}$ $\quad z \sim \mathcal{P}_z$, that is $f(f(z)) = f(z)$

Then the idempotence objective is formulated then as follows:

$$\min_\theta \delta_\theta(f_\theta(z)) = \min_\theta D\left(f_\theta(z), f_\theta(f_\theta(z))\right)$$

Does it work?

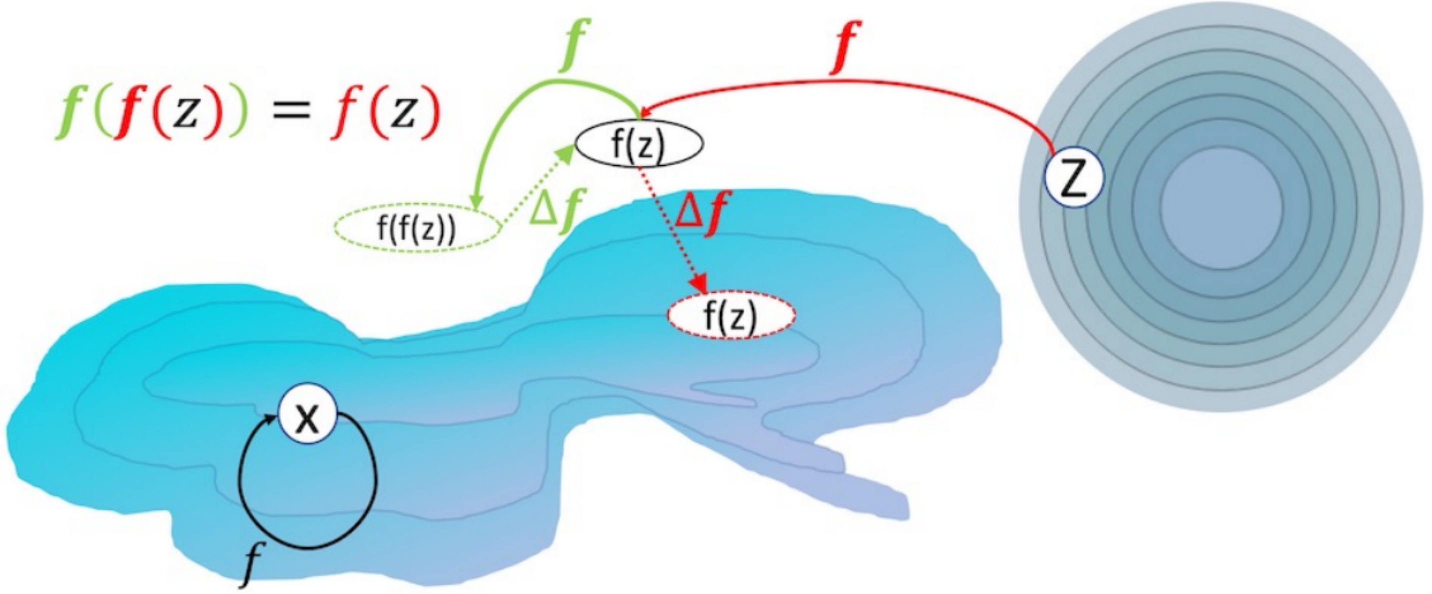
METHOD: Optimization Objectives

What about $f(z) = z \quad \forall z$?

Makes on the estimated manifold, but not imply other instances not on that

What does Idempotent optimization do?

- Mapping Z to S
- Expanding S



METHOD: Optimization Objectives

Idempotent objective

Only optimize w.r.t. the first f(·) to discourage the incentive to expand

$$L_{idem}(z;\theta,\theta') = \delta_{\theta'}(f_\theta(z)) = D\left(f_{\theta'}(f_\theta(z)), f_\theta(z)\right)$$
$$\mathcal{L}_{idem}(\theta;\theta') = \mathbb{E}_z\left[L_{idem}(z;\theta,\theta')\right]$$

Not just discourage expand, but tighten:

$$L_{tight}(z;\theta,\theta') = -\delta_\theta(f_{\theta'}(z)) = -D\left(f_\theta(f_{\theta'}(z)), f_{\theta'}(z)\right)$$

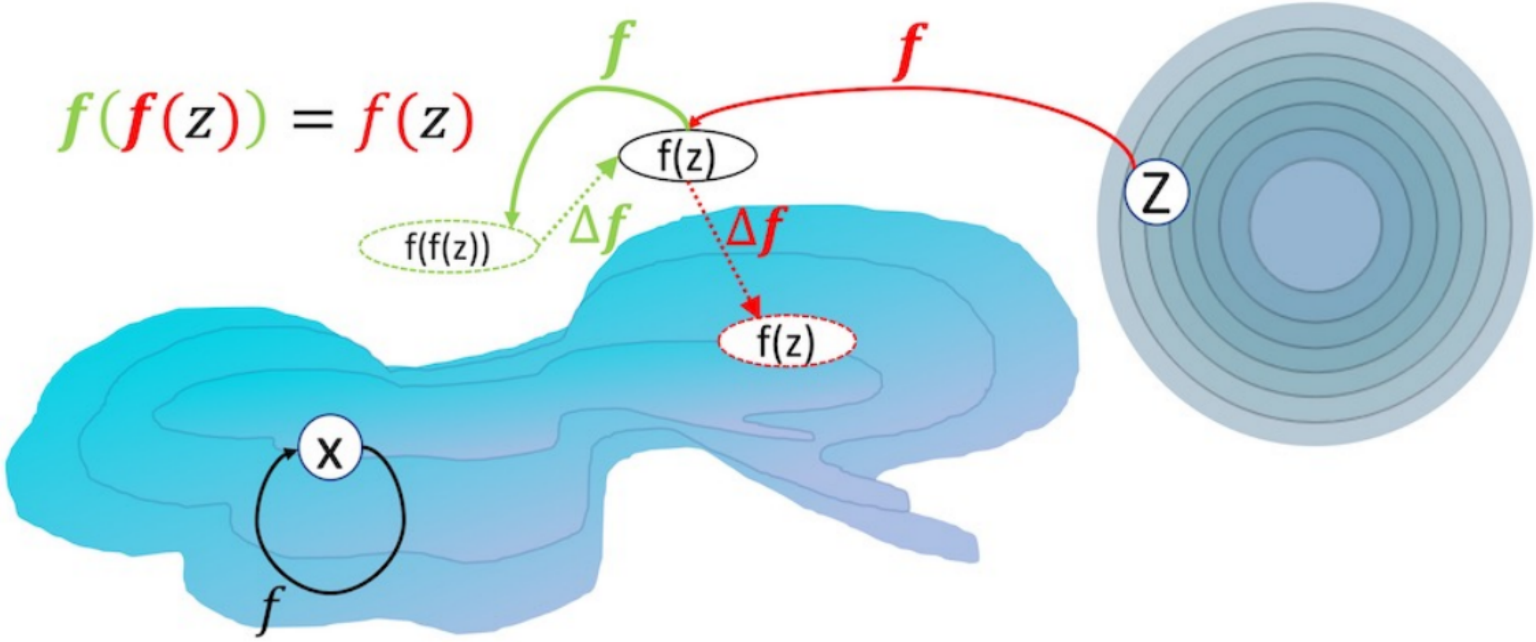
Maximize the distance between f(y) and y

METHOD: Optimization Objectives

Adversarial fashion

$$L_{tight}(z;\theta,\theta') = -L_{idem}(z;\theta',\theta)$$

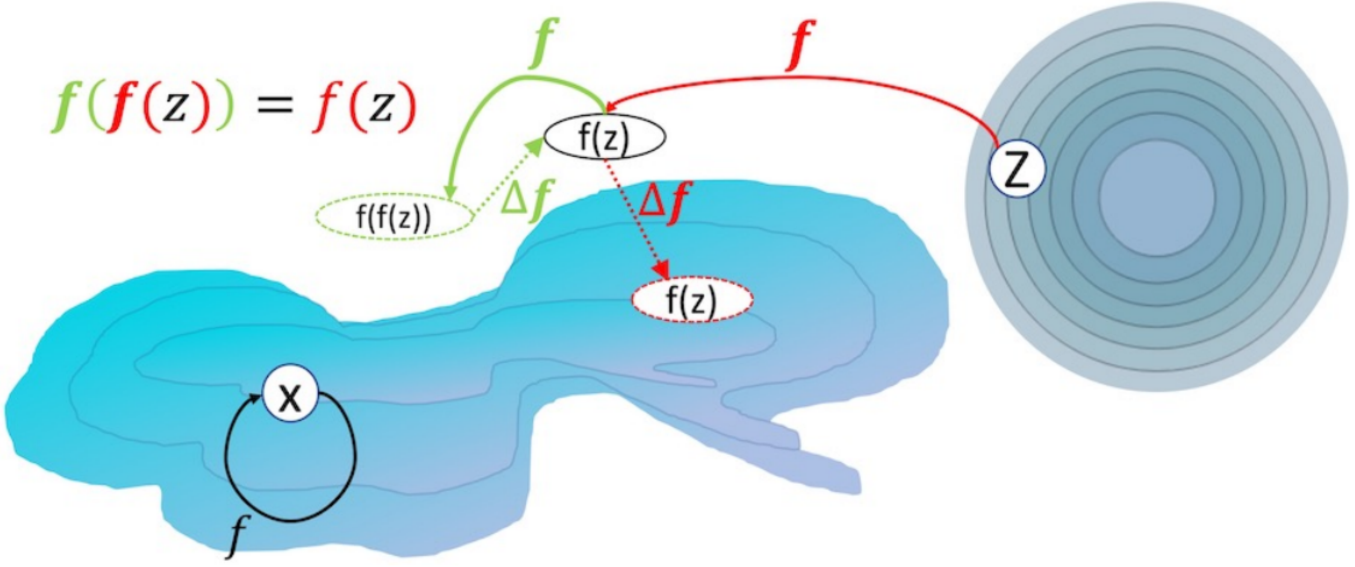
$$f(f(z)) = f(z)$$



METHOD: Optimization Objectives

Tightening loss metric

$$L_{tight}(z) = \tanh\left(\frac{\tilde{L}_{tight}(z)}{aL_{rec}(z)}\right) aL_{rec}(z)$$



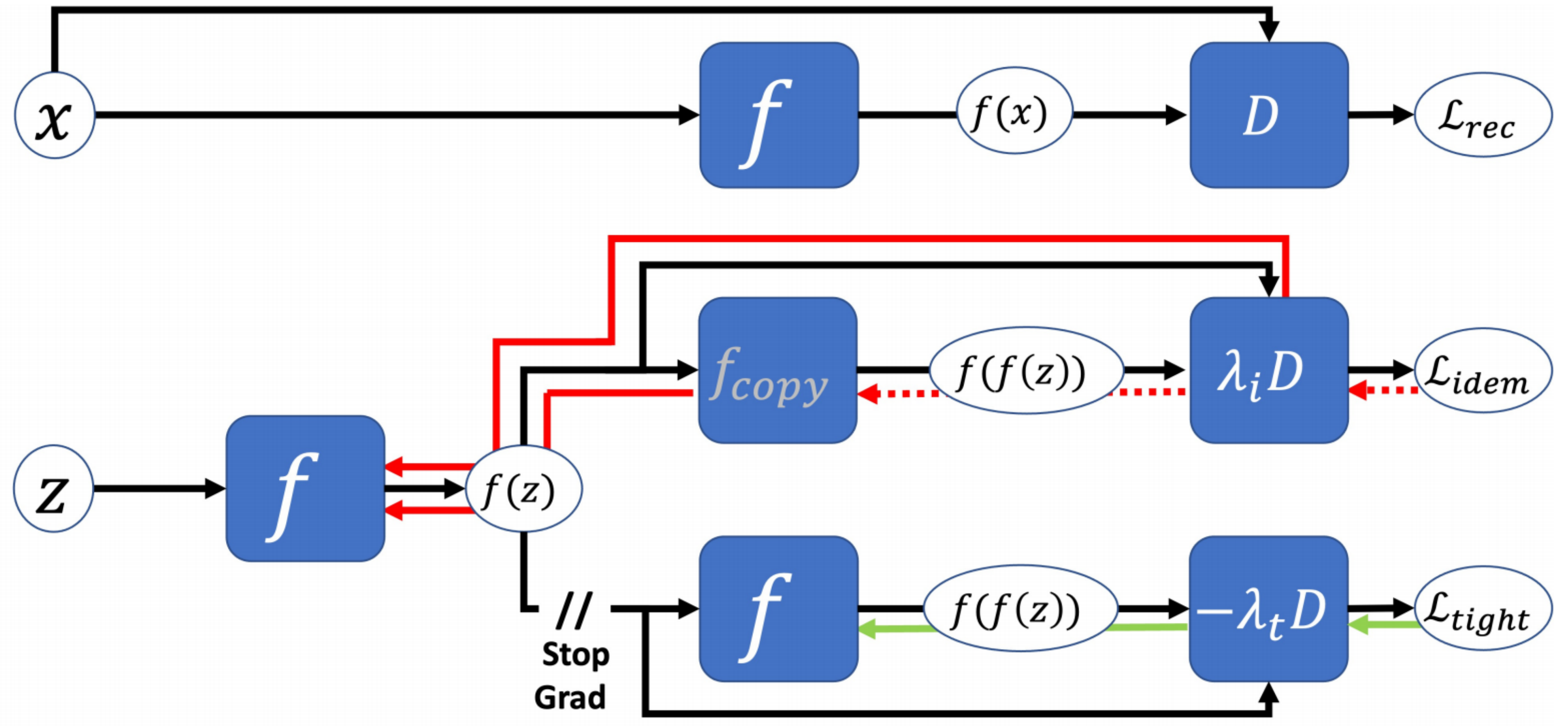
Final optimization objective

$$\begin{aligned}\mathcal{L}(\theta,\theta') &= \mathcal{L}_{rec}(\theta) + \lambda_i \mathcal{L}_{idem}(\theta;\theta') + \lambda_t \mathcal{L}_{tight}(\theta;\theta') \\ &= \mathbb{E}_{x,z}\Big[\delta_\theta(x) + \lambda_i \delta_{\theta'}(f_\theta(z)) - \lambda_t \delta_\theta(f_{\theta'}(z))\Big]\end{aligned}$$

METHOD: Training Strategy

Final optimization objective

$$\nabla_\theta \delta(f(z)) = \underbrace{\frac{\partial \delta(f(z))}{\partial f(f(z))} \frac{df(\cdot)}{d\theta}\bigg|_{f(z)}}_{\mathcal{L}_{\text{tight}}:\ \textbf{Gradient ascent}\uparrow} + \underbrace{\left(\frac{\partial \delta(f(z))}{\partial f(f(z))} \frac{\partial f(f(z))}{\partial f(z)} + \frac{\partial \delta(f(z))}{\partial f(z)}\right) \frac{df(\cdot)}{d\theta}\bigg|_{z}}_{\mathcal{L}_{\text{idem}}:\ \textbf{Gradient descent}\downarrow}$$



METHOD: Theoretical Results

How to prove the method is efficient?

- The convergenced generated distribution is aligned with target distribution

Theorem 1. *Under ideal conditions, IGN converges to the target distribution.*
We define the generated distribution, represented by $\mathcal{P}_\theta(y)$, as the PDF of y when $y = f_\theta(z)$ and $z \sim \mathcal{P}_z$. We split the loss into two terms.

$$\mathcal{L}(\theta;\theta') = \underbrace{\mathcal{L}_{rec}(\theta) + \lambda_i \mathcal{L}_{tight}(\theta;\theta')}_{\mathcal{L}_{rt}} + \lambda_t \mathcal{L}_{idem}(\theta;\theta') \tag{15}$$

We assume a large enough model capacity such that both terms obtain a global minimum:

$$\theta^* = \arg\min_\theta \mathcal{L}_{rt}(\theta;\theta^*) = \arg\min_\theta \mathcal{L}_{idem}(\theta;\theta^*) \tag{16}$$

Then, $\exists \theta^* : \mathcal{P}_{\theta^*} = \mathcal{P}_x$ and for $\lambda_t = 1$, this is the only possible $\mathcal{P}_{\theta^*}$.

METHOD: Theoretical Results

We first find the global minimum of $\mathcal{L}_{rt}$ given the current parameters θ^*:

$$\mathcal{L}_{rt}(\theta;\theta^*) = \mathbb{E}_x\left[D(f_\theta(x), x)\right] - \lambda_t \mathbb{E}_z\left[D(f_\theta(f_{\theta^*}(z)), f_{\theta^*}(z))\right] \tag{17}$$

$$= \int \delta_\theta(x)\mathcal{P}_x(x)dx - \lambda_t \int \delta_\theta(f_{\theta^*}(z))\mathcal{P}_{\theta^*}(z)dz \tag{18}$$

We now change variables. For the left integral, let $y := x$ and for the right integral, let $y := f_{\theta^*}(z)$.

$$\mathcal{L}_{rt}(\theta;\theta^*) = \int \delta_\theta(y)\mathcal{P}_x(y)dy - \lambda_t \int \delta_\theta(y)\mathcal{P}_{\theta^*}(y)dy \tag{19}$$

$$= \int \delta_\theta(y)\Big(\mathcal{P}_x(y) - \lambda_t \mathcal{P}_{\theta^*}(y)\Big)dy \tag{20}$$

METHOD: Theoretical Results

We denote $M = \sup_{y_1, y_2} D(y_1, y_2)$, where the supremum is taken over all possible pairs y_1, y_2. Note that M can be infinity. Since δ_θ is non-negative, the global minimum for $\mathcal{L}_{rt}(\theta; \theta^*)$ is obtained when:

$$\delta_{\theta^*}(y) = M \cdot \mathbb{1}_{\{\mathcal{P}_x(y) < \lambda_t \mathcal{P}_{\theta^*}(y)\}} \quad \forall y \tag{21}$$

Next, we characterize the global minimum of $\mathcal{L}_{idem}$ given the current parameters θ^*:

$$\mathcal{L}_{idem}(\theta, \theta^*) = \mathbb{E}_z \left[D\left(f_{\theta^*}(f_\theta(z)), f_\theta(z) \right) \right] = \mathbb{E}_z \left[\delta_{\theta^*}(f_\theta(z)) \right] \tag{22}$$

Plugging in Eq. 21 and substituting θ^* with θ as we examine the minimum of the inner f:

$$\mathcal{L}_{idem}(\theta; \theta^*) = M \cdot \mathbb{E}_z \left[\mathbb{1}_{\{\mathcal{P}_x(y) < \lambda_t \mathcal{P}_\theta(y)\}} \right] \tag{23}$$

METHOD: Theoretical Results

To obtain θ^*, according to our assumption in Eq. 16, we take $\arg\min_\theta$ of Eq. 23:

$$\theta^* = M \cdot \arg\min_\theta \mathbb{E}_z \left[\mathbb{1}_{\{\mathcal{P}_x(y) < \lambda_t \mathcal{P}_\theta(y)\}} \right] \tag{24}$$

The presence of parameters to be optimized in this formulation is in the notion of the distribution $\mathcal{P}_\theta(y)$. If $\mathcal{P}_{\theta*} = \mathcal{P}_x$ and $\lambda_t \leq 1$, the loss value will be 0, which is its minimum. If $\lambda = 1$, $\theta^* : \mathcal{P}_{\theta*} = \mathcal{P}_x$ is the only minimizer. This is because the total sum of the probability needs to be 1. Any y for which $\mathcal{P}_\theta(y) < \mathcal{P}_x(y)$ would necessarily imply that $\exists y$ such that $\mathcal{P}_\theta(y) > \mathcal{P}_x(y)$, which would increase the loss. $\square$

METHOD: Theoretical Results

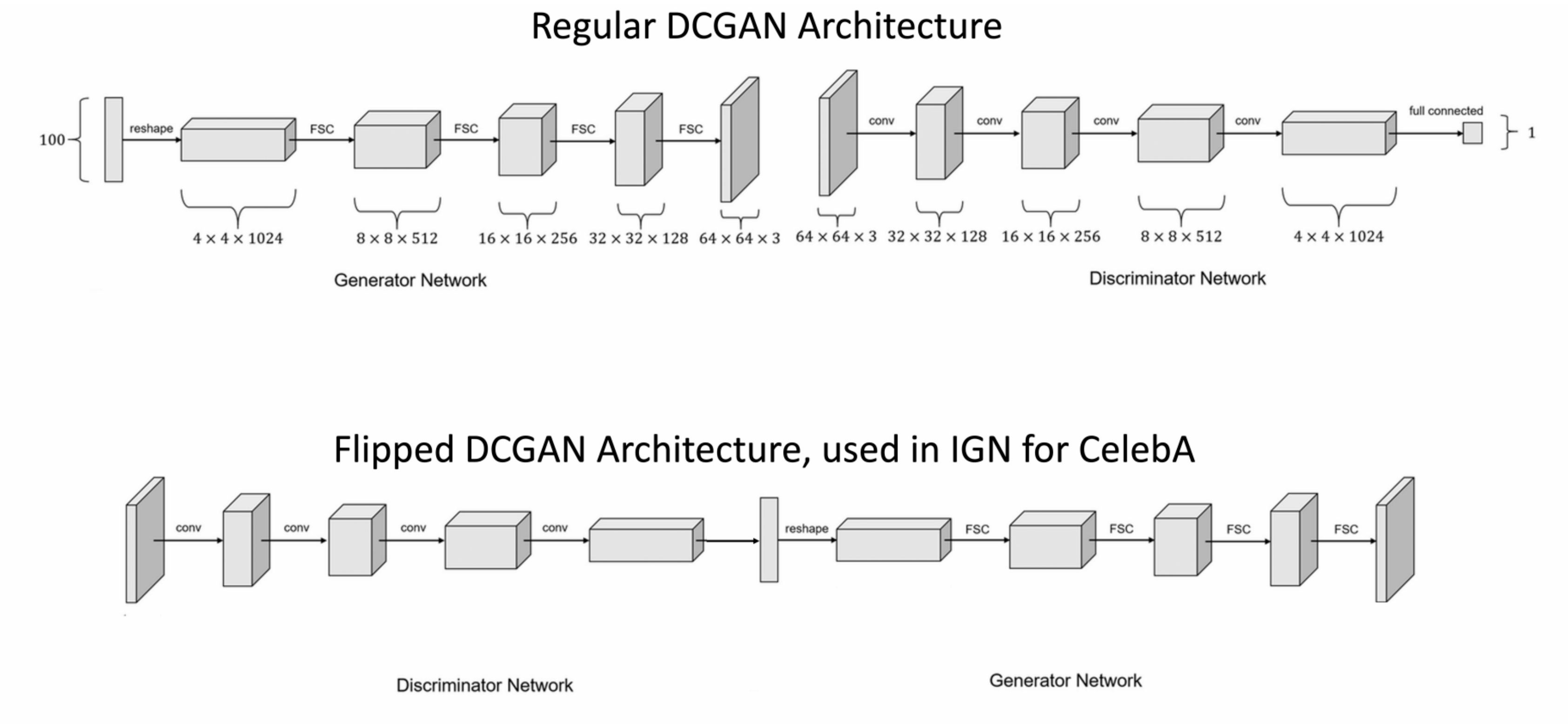
In practice, we use $\lambda_t < 1$. While the theoretical derivation guarantees a single desired optimum for $\lambda_t = 1$, the practical optimization of a finite capacity neural network suffers undesirable effects such as instability. The fact that f is continuous makes the optimal theoretical θ^* which produces a discontinuous δ_{θ^*} unobtainable in practice. This means that $\mathcal{L}_{tight}$ tends to push toward high values of $\delta_\theta(y)$ also for y that is in the estimated manifold. Moreover, in general, it is easier to maximize distances than minimize them, just by getting big gradient values.

OUTLINE

- Background
- Method
- Experiments
- Conclusion

EXPERIMENTS

Network architecture: Autoencoder from DCGAN

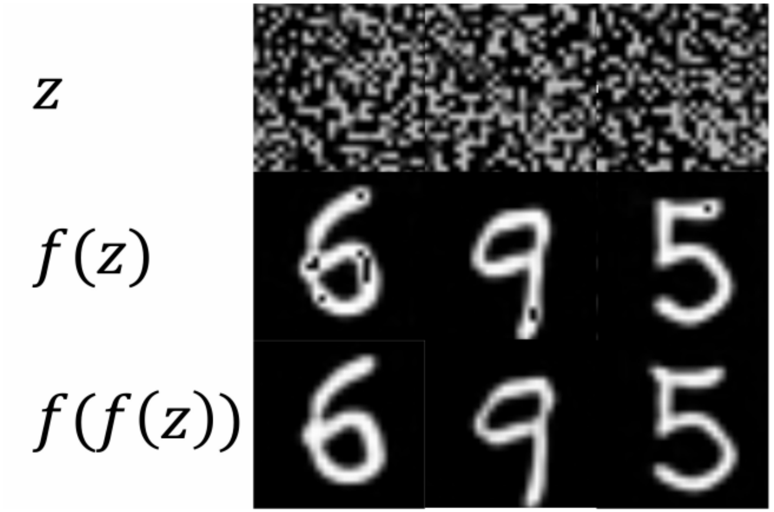


Dataset: MNIST(28*28) , CelebA(64*64)

EXPERIMENTS

Generation from noise

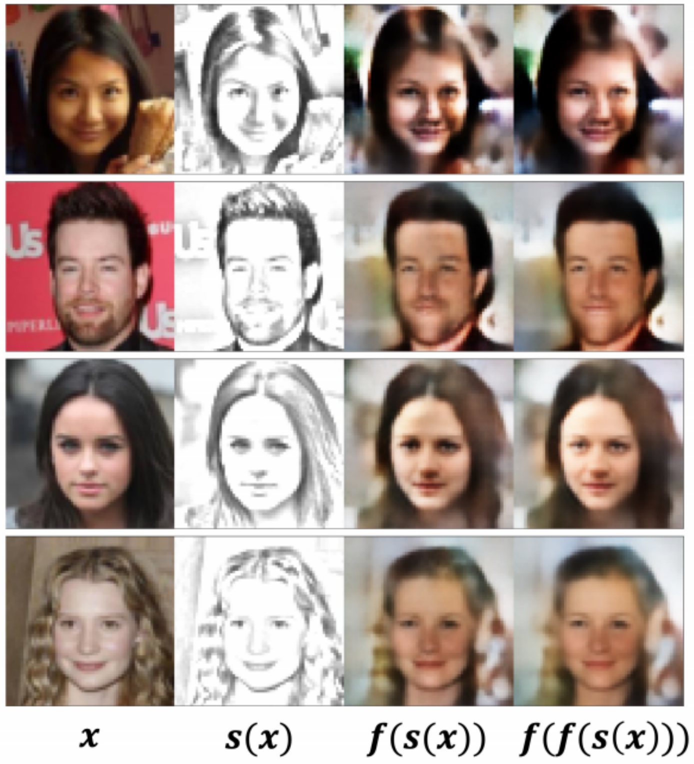
FID=39 (DCGAN FID=34)



EXPERIMENTS

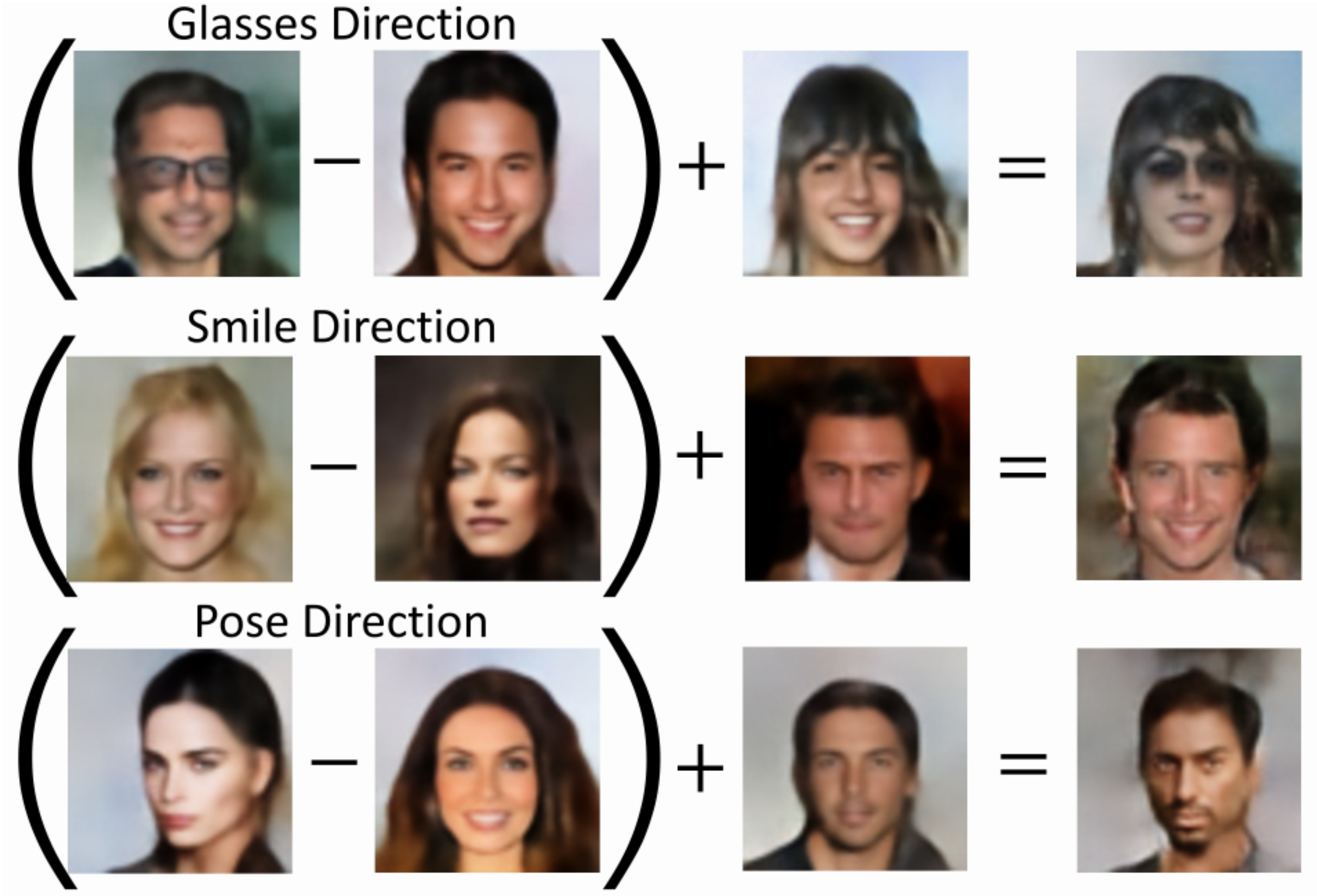
Out-of-distribution projection

Generation from noisy image, grayscale and sketches



EXPERIMENTS

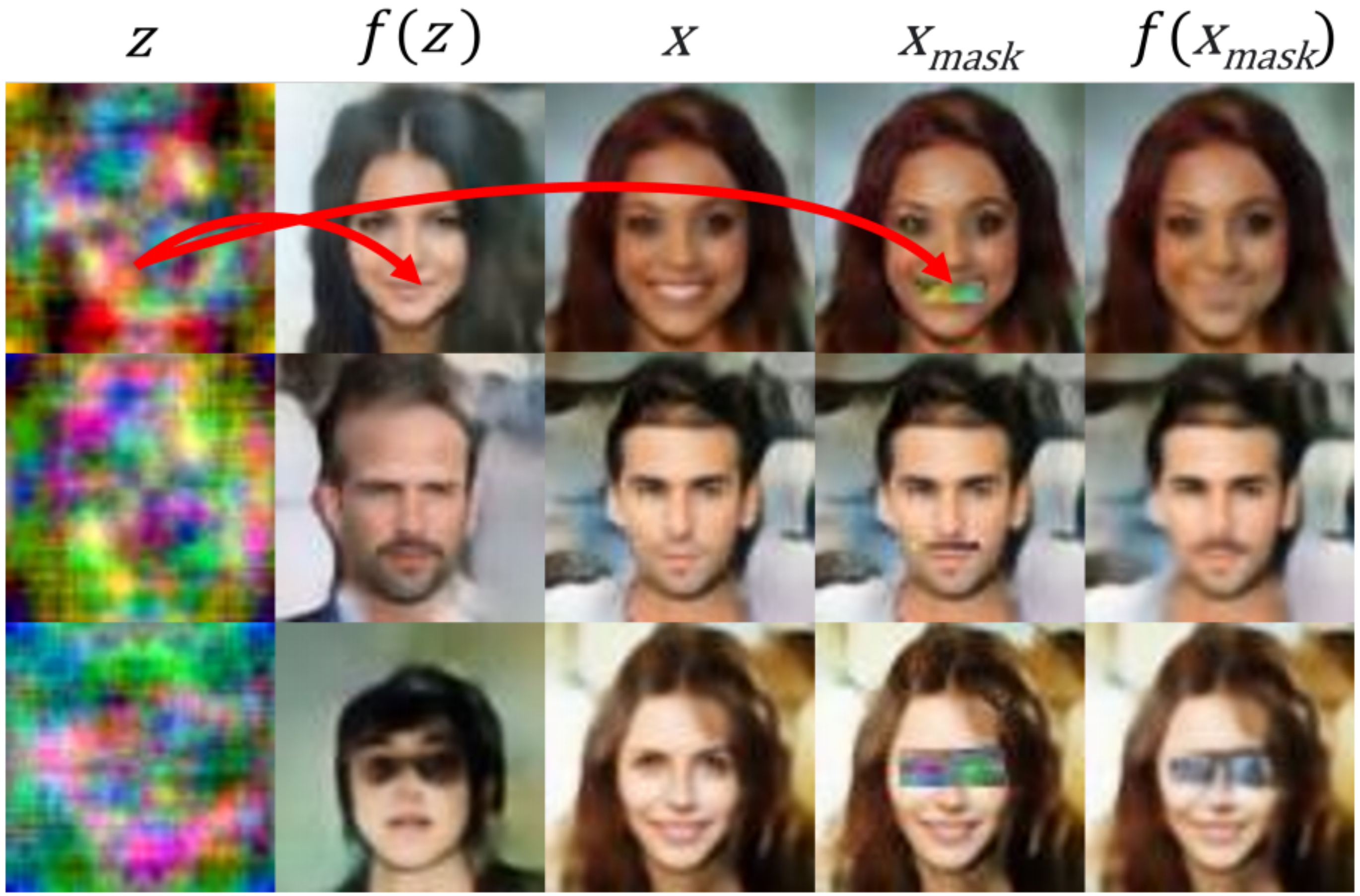
Latent space manipulations



EXPERIMENTS

Projection-based edit

Projection-based compositing



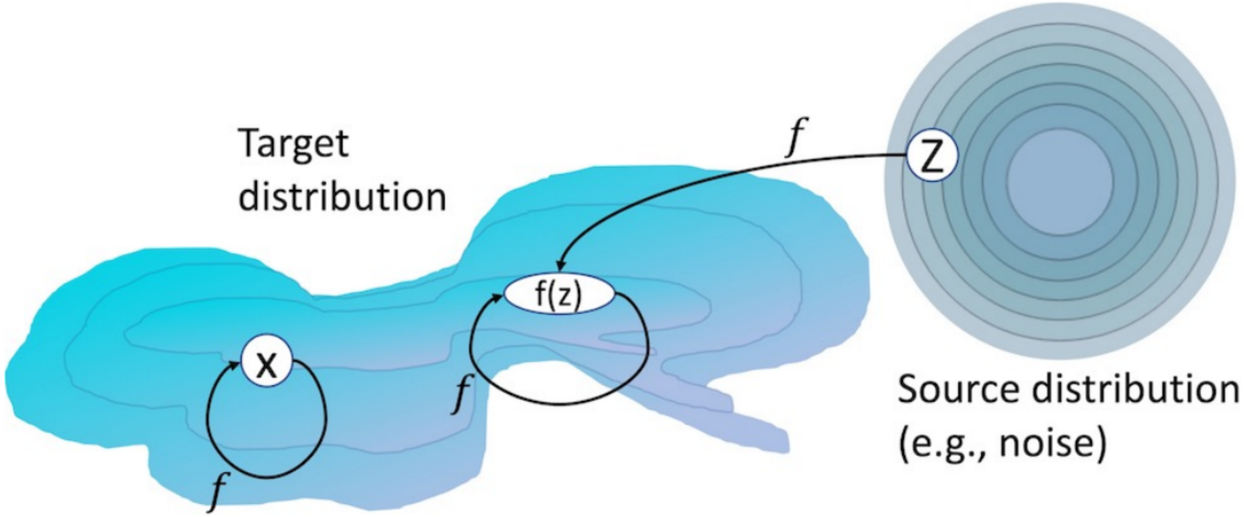
OUTLINE

- Background
- Method
- Experiments
- Conclusion

CONCLUSION

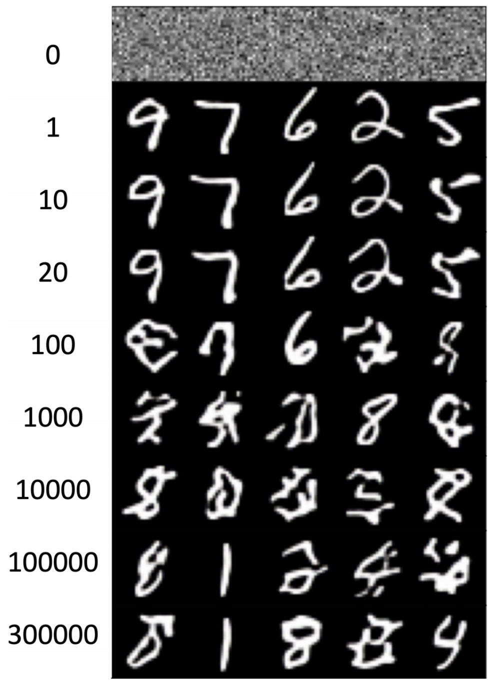
- Compared with GAN
  - Self-adversarial
- Compared with Diffusion
  - The trajectory between distributions is determined solely by the model's learning process but not set rule

$$f(x) = x$$
$$f(f(z)) = f(z)$$



CONCLUSION

- **Advantage**
  - A global projector, can apply to never seen data
  - One step projector
  - Allow more accurate finetune by multi-step map
- **Limitation**
  - Mode collapse
  - Blurriness
  - Unsteadiness



Thanks for listening!