Idempotet Generative Network

Assaf Shocher[1,2], Amil Dravid[1] Yossi Gandelsman[1]
Inbar Mosseri[2] Michael Rubinstein[2] Alexei A. Efros[1]

[1] UC Berkeley
[2] Google Research

ICLR 2024

Make It Real

The Idempotent Generative Network (IGN) aims to create a model that handles diverse inputs and projects them onto the target distribution in one step.

Unlike GANs or VAEs, which need specific inputs, IGN acts as a “global projector.” It turns any input—corrupted images, sketches, or noise—into real images instantly, like a “Make It Real” button.

x $\quad x+n$ $\quad f(x+n)$ $\quad f(f(x+n))$

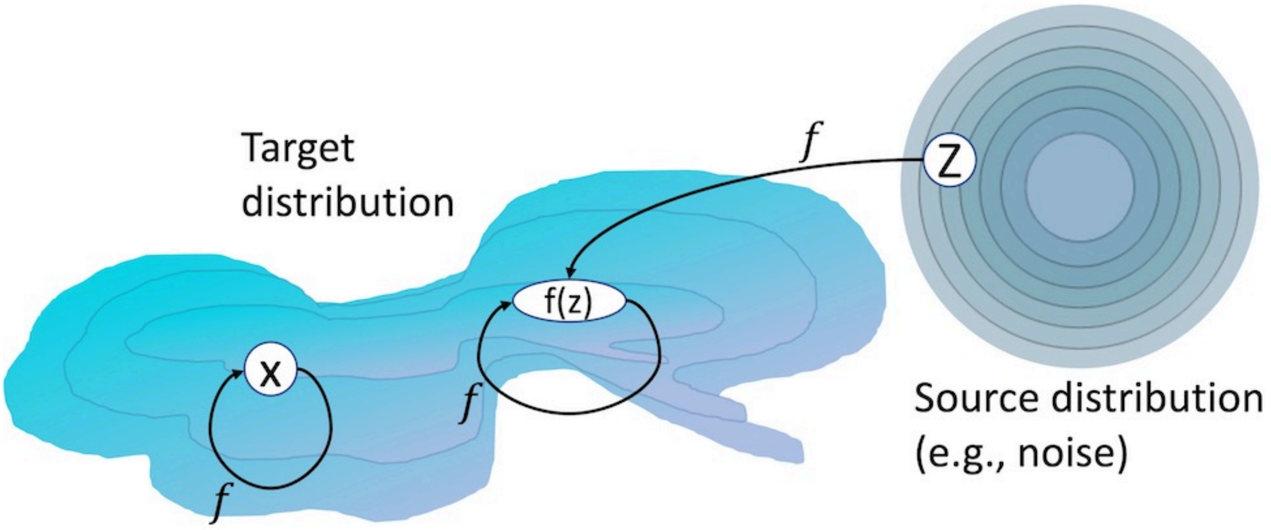
x $\quad g(x)$ $\quad f(g(x))$ $\quad f(f(g(x)))$

x $\quad m(x)$ $\quad f(m(x))$ $\quad f(f(m(x)))$

Idempotency: One Step Generation, Optional Refinement

Disadvantages and Problems

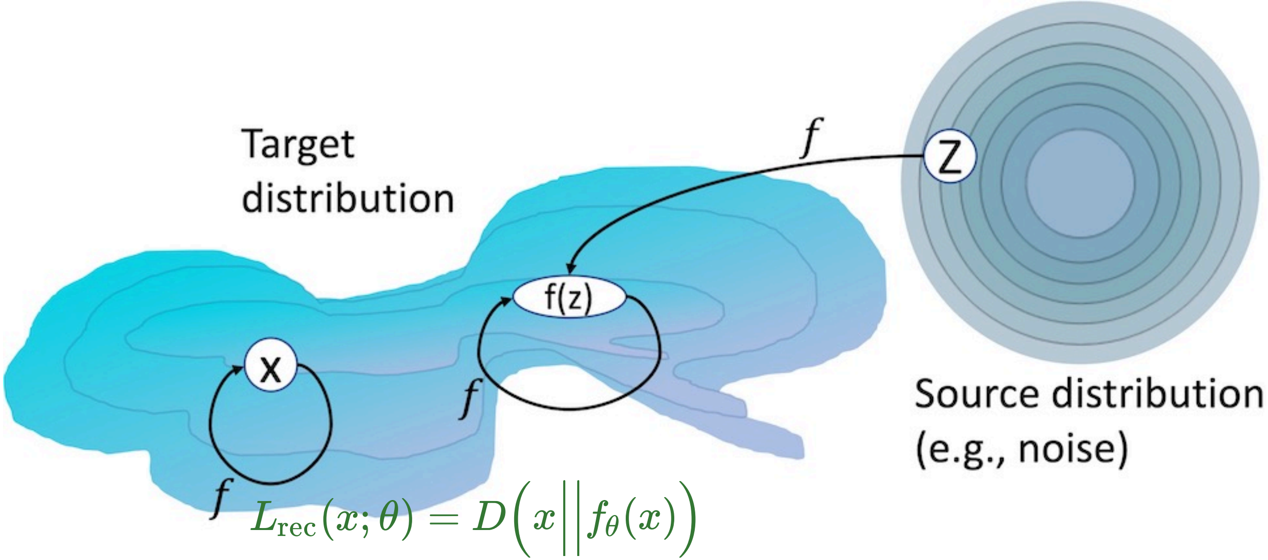
IGN is built on idempotency, where its operator f ensures $f(f(z)) = f(z)$ and $f(x) = x$ for target data. It generates outputs in one step like GANs, with optional refinement akin to diffusion models. Unlike those, IGN keeps a consistent latent space for easy manipulation.



Idempotency: One Step Generation, Optional Refinement

Disadvantages and Problems

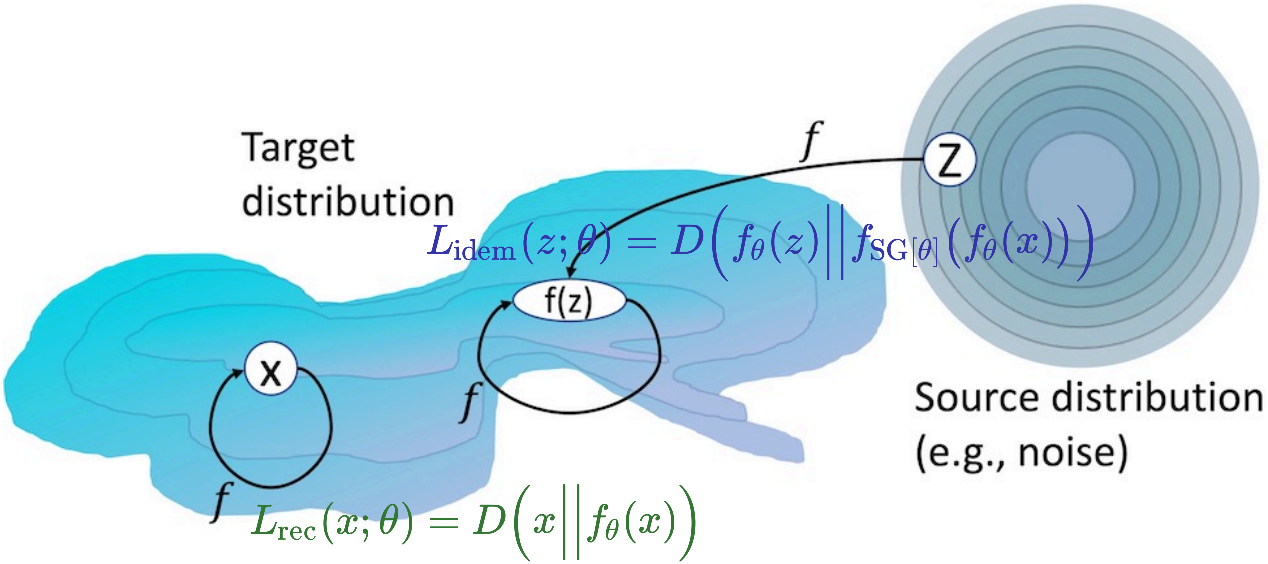
IGN is built on idempotency, where its operator f ensures $f(f(z)) = f(z)$ and $f(x) = x$ for target data. It generates outputs in one step like GANs, with optional refinement akin to diffusion models. Unlike those, IGN keeps a consistent latent space for easy manipulation.

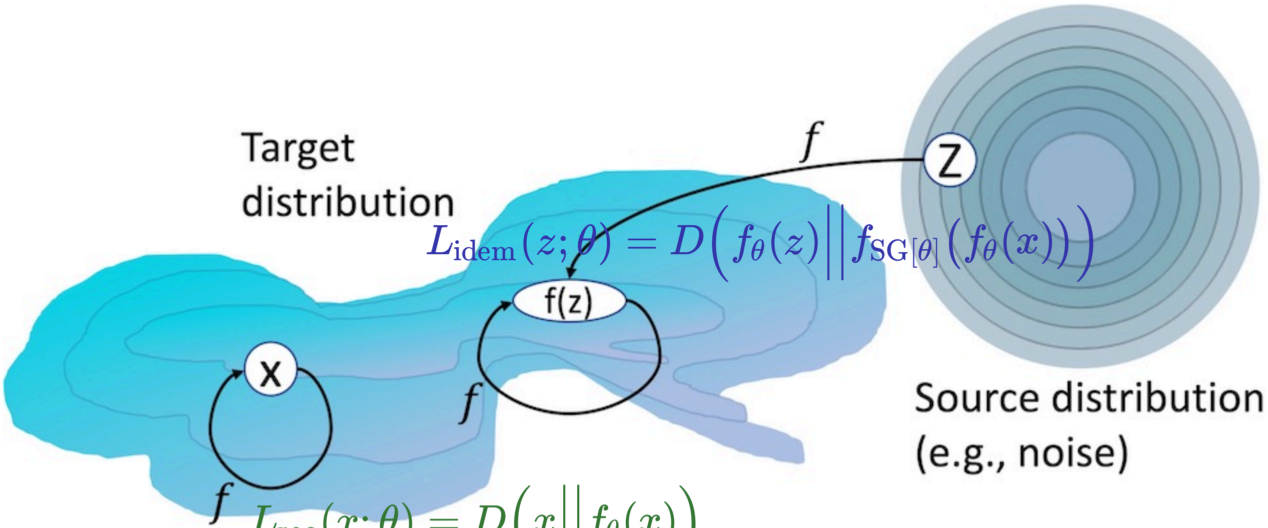


Idempotency: One Step Generation, Optional Refinement

Disadvantages and Problems

IGN is built on idempotency, where its operator f ensures $f(f(z)) = f(z)$ and $f(x) = x$ for target data. It generates outputs in one step like GANs, with optional refinement akin to diffusion models. Unlike those, IGN keeps a consistent latent space for easy manipulation.

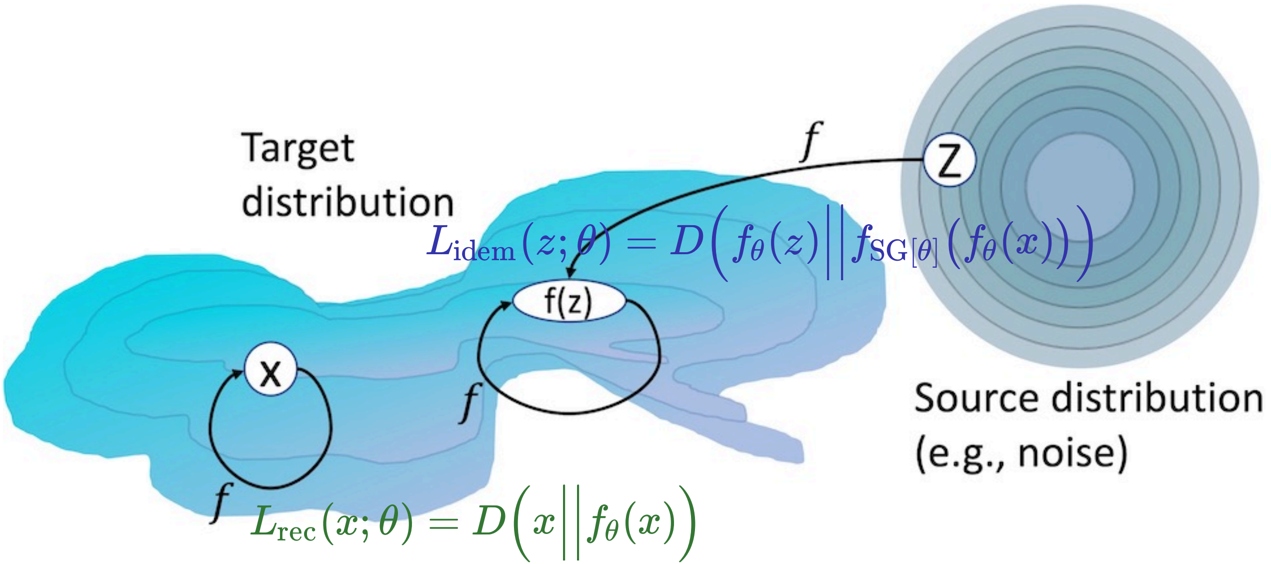




$$L_{\text{idem}}(z;\theta) = D\Big(f_\theta(z)\Big|\Big|f_{\text{SG}[\theta]}\big(f_\theta(x)\big)\Big)$$

$$L_{\text{rec}}(x;\theta) = D\Big(x\Big|\Big|f_\theta(x)\Big)$$

However, the combination of these two loss functions by itself does not guarantee that the target manifold is compact and contains only the true data distribution.

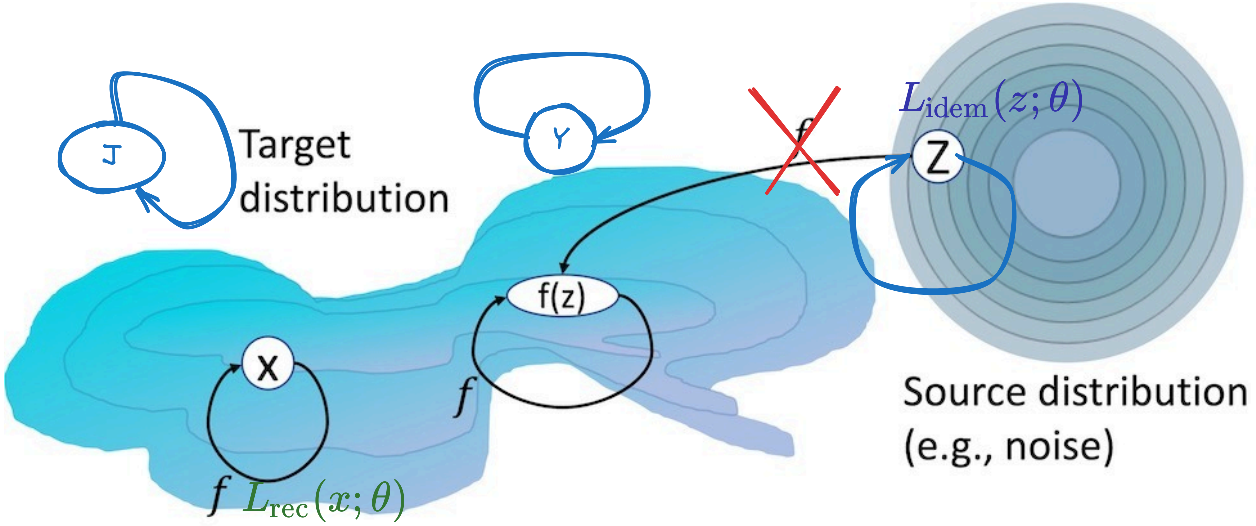


However, the combination of these two loss functions by itself does not guarantee that the target manifold is compact and contains only the true data distribution.

The Issue of Manifold Expansion

The Issue of Manifold Expansion

If the model f directly becomes the identity function, that is, $f(z) = z, \forall z$. Both L_{rec} and L_{idem} are perfectly satisfied. However, this model does **NOT LEARN** to generate any new samples that **RESEMBLE THE TARGET DISTRIBUTION**.



Tightness Objective

$$\tilde{L}_{\text{tight}}(z;\theta) = -D\left(f_{\text{SG}[\theta]}(z) \middle\| f_\theta(f_{\text{SG}[\theta]}(z))\right)$$

Tightness Objective: Self-Adversarial Training

$$\tilde{L}_{\text{tight}}(z;\theta) = -D\Big(f_{\text{SG}[\theta]}(z)\Big|\Big|f_\theta\big(f_{\text{SG}[\theta]}(z)\big)\Big)$$

$$\Updownarrow$$

$$L_{\text{idem}}(z;\theta) = D\Big(f_\theta(z)\Big|\Big|f_{\text{SG}[\theta]}\big(f_\theta(x)\big)\Big)$$

Tightness Objective: Self-Adversarial Training

$$\tilde{L}_{\text{tight}}(z;\theta) = -D\Big(f_{\text{SG}[\theta]}(z)\Big|\Big|f_\theta\big(f_{\text{SG}[\theta]}(z)\big)\Big)$$

$$\Updownarrow$$

$$L_{\text{idem}}(z;\theta) = D\Big(f_\theta(z)\Big|\Big|f_{\text{SG}[\theta]}\big(f_\theta(x)\big)\Big)$$

However, optimizing directly against the tightness objective will conflict with the generation target.

Tightness Objective: Self-Adversarial Training

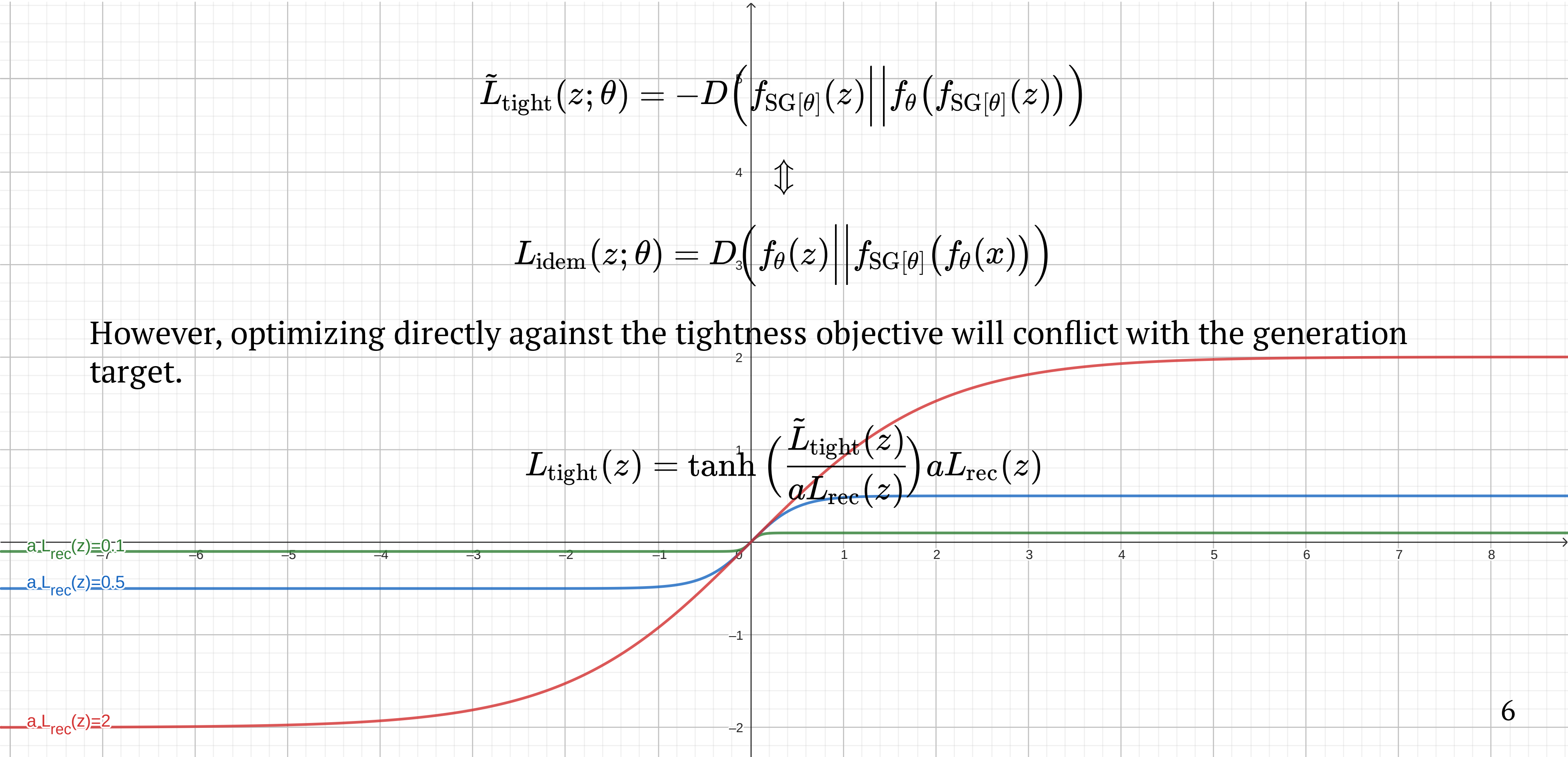
$$\tilde{L}_{\text{tight}}(z;\theta) = -D\Big(f_{\text{SG}[\theta]}(z)\Big|\Big|f_\theta\big(f_{\text{SG}[\theta]}(z)\big)\Big)$$

$$\Updownarrow$$

$$L_{\text{idem}}(z;\theta) = D\Big(f_\theta(z)\Big|\Big|f_{\text{SG}[\theta]}\big(f_\theta(x)\big)\Big)$$

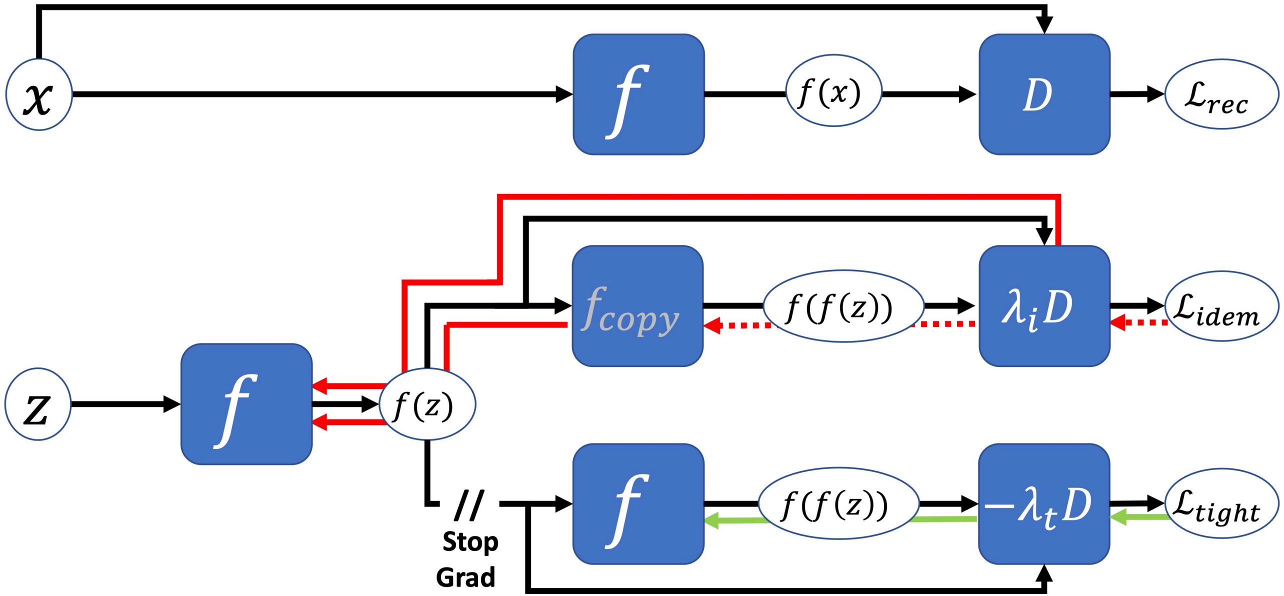
However, optimizing directly against the tightness objective will conflict with the generation target.

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

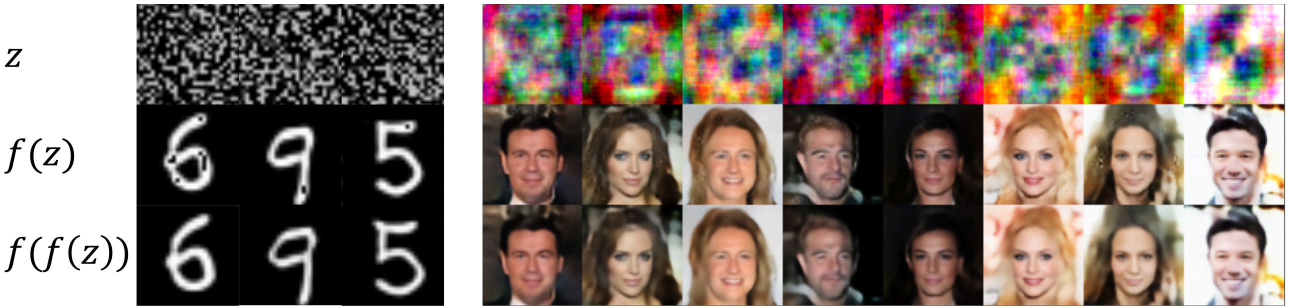


Training Objective

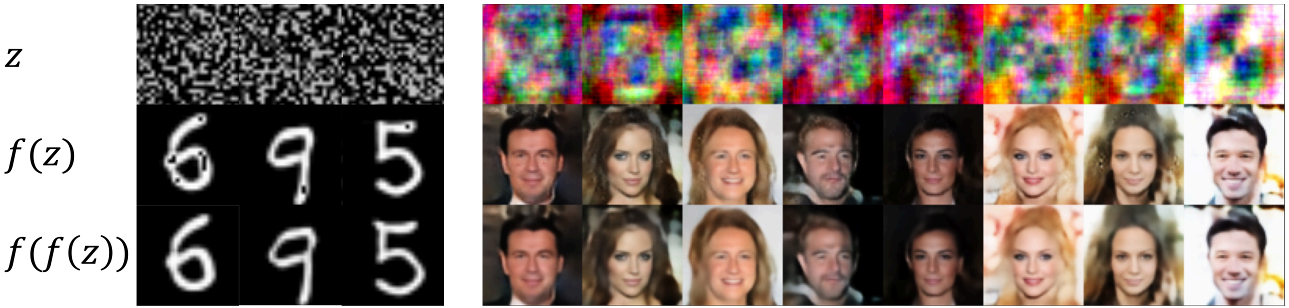
$$\mathcal{L}(\theta) = \lambda_r \mathbb{E}_x [L_{\text{rec}}(x;\theta)] + \lambda_i \mathbb{E}_z [L_{\text{idem}}(z;\theta)] + \lambda_t \mathbb{E}_z [L_{\text{tight}}(z;\theta)]$$



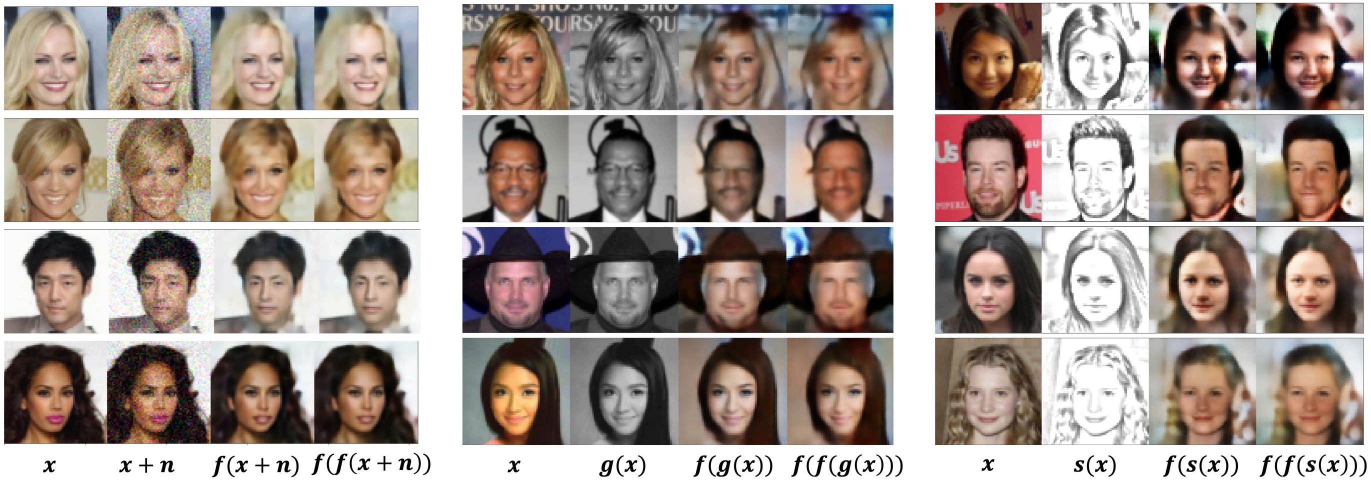
Experiment



Experiment

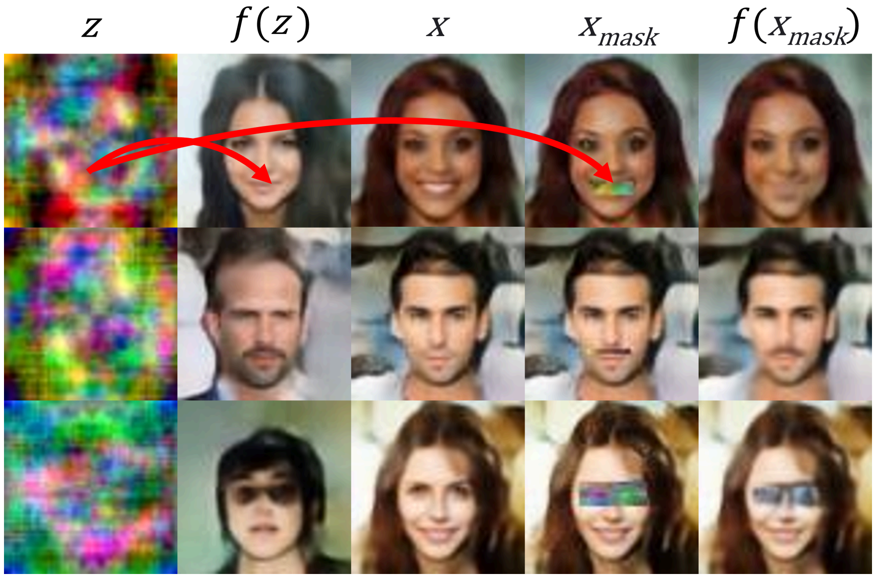


Make It Real



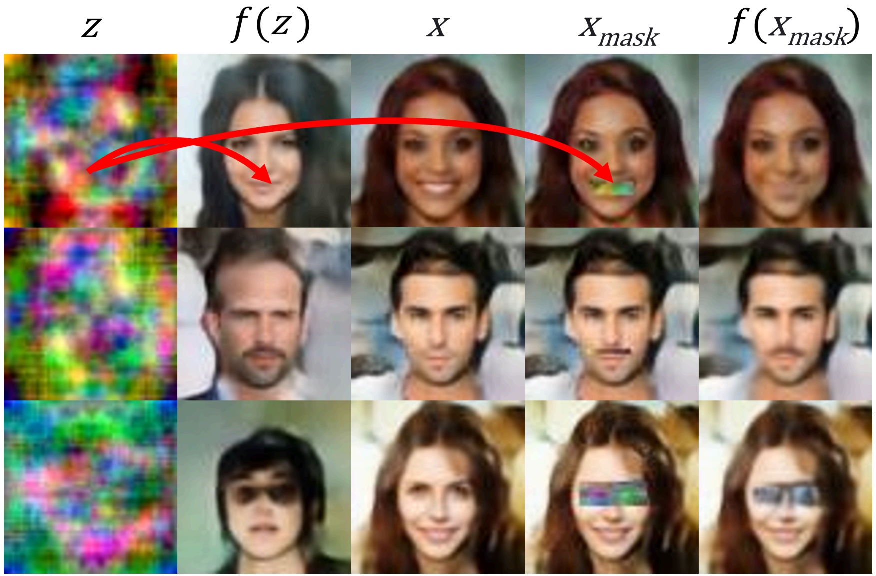
Experiment

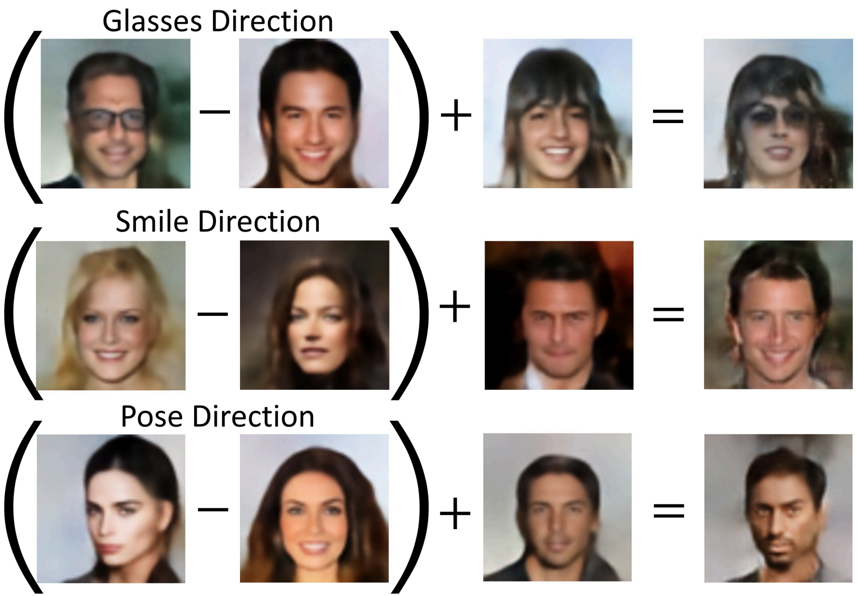
Latent Space Manipulations.



Experiment

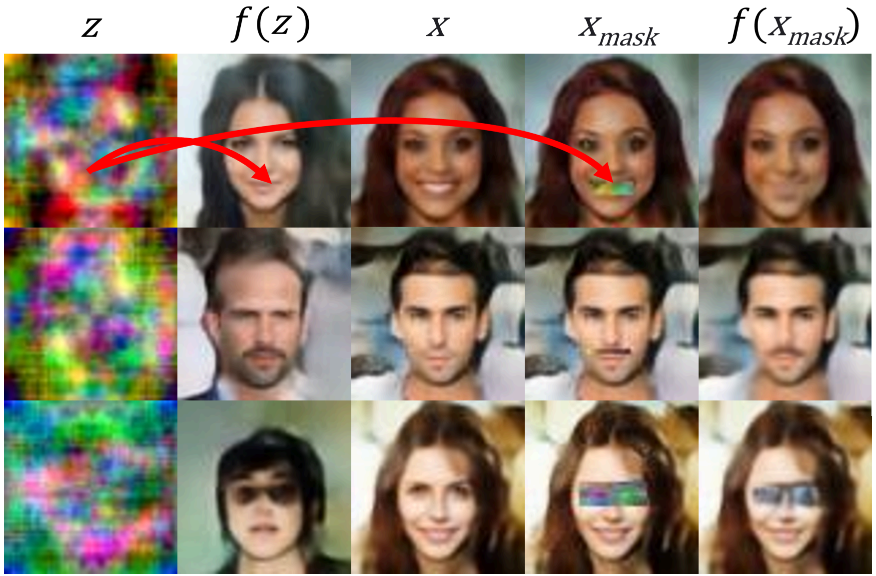
Latent Space Manipulations.

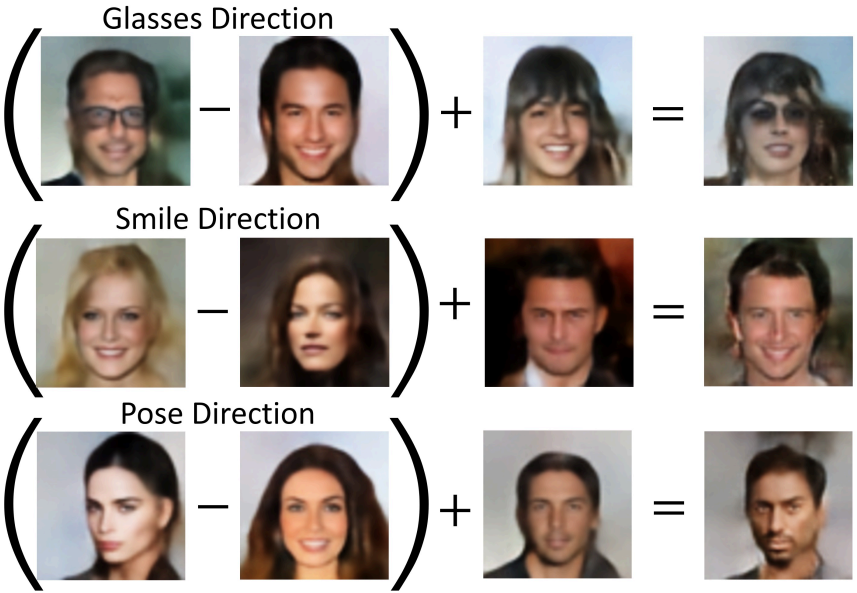




Experiment

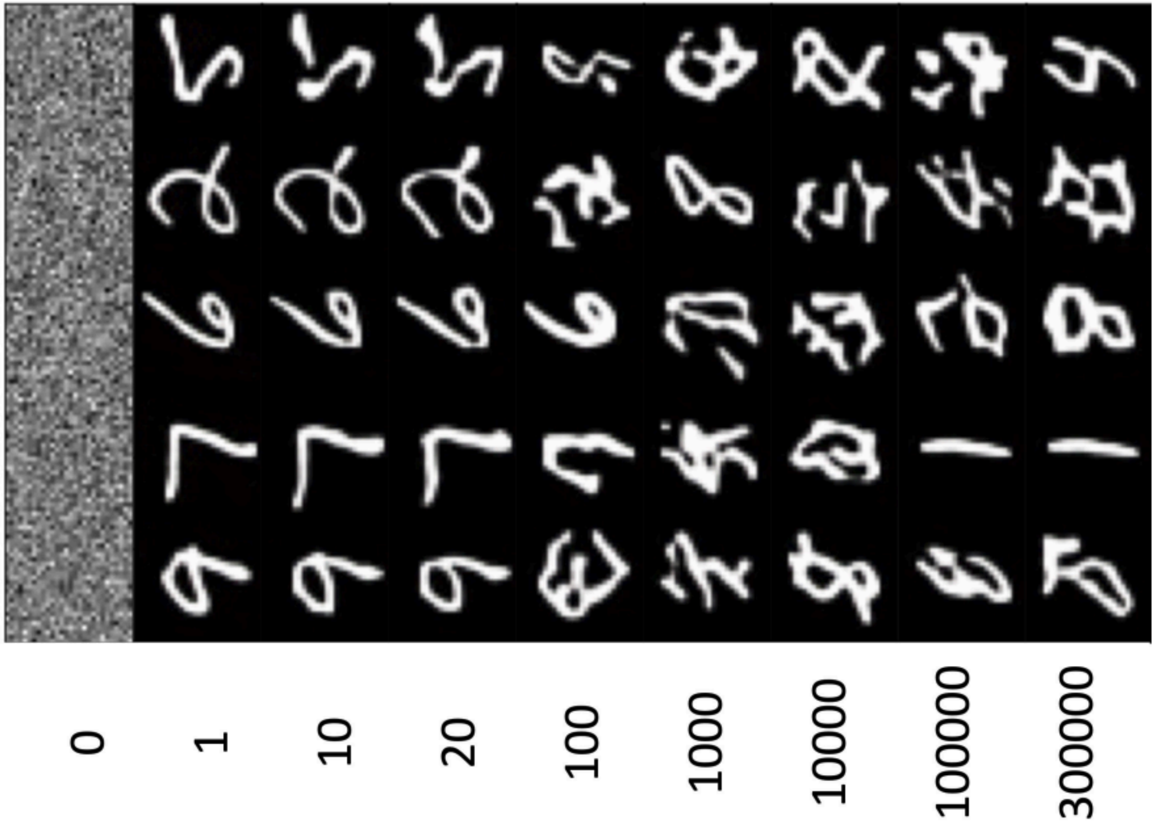
Latent Space Manipulations.





Experiment

$f^{k \to \infty}(z)$



Conclusions

Advantages

- Single-step generation is fast and efficient.
- Optional sequence refinement.
- Maintains consistent latent space.
- Good generalization and projection capabilities.
- Simplifies editing process.

Weaknesses

- Generation quality needs to be improved.
- Mode collapse may occur.
- Potential issue tightness objective.
  It may impose large modifications on relatively good generated instances and may encourage high gradients and instability.

END