Score-based Idempotent Distillation of Diffusion Models

Shehtab Zaman[*], **Chengyan Liu**[*], **& Kenneth Chiu**
School of Computing
State University of New York at Binghamton
Binghamton, NY 13902, USA
{szaman5,cliu118,kchiu}@binghamton.edu

Abstract

Diffusion and score-based models are popular approaches to generative modeling that iteratively transport samples from one distribution, usually a Gaussian, to a target data distribution. These models have gained popularity due to their stable training dynamics and high-fidelity generation quality. However, this stability and quality come at the cost of high computational cost, as the data must be transported incrementally along the entire trajectory. New sampling methods, model distillation, and consistency models have been developed to reduce the sampling cost and even perform one-shot sampling from diffusion models. Idempotent generative networks (IGNs), by contrast, project samples to a data manifold in a single operation. In this work, we unite diffusion and idempotent models by training idempotent models through distillation from diffusion models. We present Score-based Idempotent Generative Networks (SIGNs), generative models that support both single- and multi-step generation. We show that idempotent networks, under the idempotent constraint, can effectively distill a pre-trained diffusion model and enable faster inference compared to iterative score-based models. Like IGNs and score-based models, SIGNs can perform multi-step sampling, allowing users to trade off quality for efficiency. As these models operate directly on the source domain, they can project corrupted or alternate distributions back onto the target manifold, enabling zero-shot editing of inputs. We validate our models on a simple multi-modal dataset as well as multiple image datasets, achieving state-of-the-art results for idempotent models on the CIFAR and CelebA datasets.

1 Introduction

Generative modeling for high-dimensional data like images and video faces a fundamental trilemma Xiao et al. [2021]: balancing (a) high sample quality, (b) fast sampling, and (c) diverse mode coverage Yu et al. [2020], Zhao et al. [2018]. This challenge has driven the development of numerous deep generative methods, each navigating these trade-offs differently. Prominent examples include generative adversarial networks (GANs) Goodfellow et al. [2020], variational autoencoders (VAEs) Kingma and Welling [2013], auto-regressive models Van Den Oord et al. [2016], normalizing flows Rezende and Mohamed [2015], Dinh et al. [2014], Ho et al. [2019], flow-matching, and diffusion models Sohl-Dickstein et al. [2015], Ho et al. [2020], Nichol and Dhariwal [2021].

GANs generate high-quality samples quickly but suffer from training instabilities and reduced mode coverage due to their adversarial objective, which can lead to mode collapse Salimans et al. [2016], Kodali et al. [2017], Jo et al. [2020]. Also, capable of single-step sampling, normalizing flows and VAEs often produce lower-quality samples Ho et al. [2022a].

[*]Equal Contribution. Corresponding author: szaman5@binghamton.edu

accepted at third SPIGM workshop @ NeurIPS 2025

Idempotent Generative Models (IGNs) Shocher et al. [2023] are the newest entry in the zoo of generative models. They are a novel class of GAN-like models that can combine the benefits of both diffusion models and GANs. They support single-step sampling and iterative refinement, offering a flexible trade-off between computational cost and sample quality. Like GANs, IGNs suffer from training instabilities stemming from their adversarial objective. Modern generative model training requires a large amount of data and computing resources. Models with unstable training, where the model can abruptly diverge, are prohibitively costly to train. Improving the training stability of IGNs is crucial for effective design exploration and for enhancing model performance.

Diffusion and score-matching models have become the de facto generative models. They achieve high sample quality and are significantly easier to optimize. This training stability has enabled researchers to rigorously optimize these models, leading to architectural innovations and state-of-the-art performance in domains such as image, video Ho et al. [2022b], Mei and Patel [2023], audio generation Kong et al. [2020], molecular synthesis Schneuing et al. [2022], Hoogeboom et al. [2022], Xu et al. [2022], and protein structure prediction Corso et al. [2022]. However, this performance comes at the cost of slow, iterative sampling, as these models require multiple steps to transform a sample from a simple prior distribution to the complex data distribution. To address this limitation, many recent works like Song and Dhariwal [2023], Xiao et al. [2021], Sauer et al. [2023] have focused on accelerating sampling through methods like distillation, without sacrificing model quality or stability.

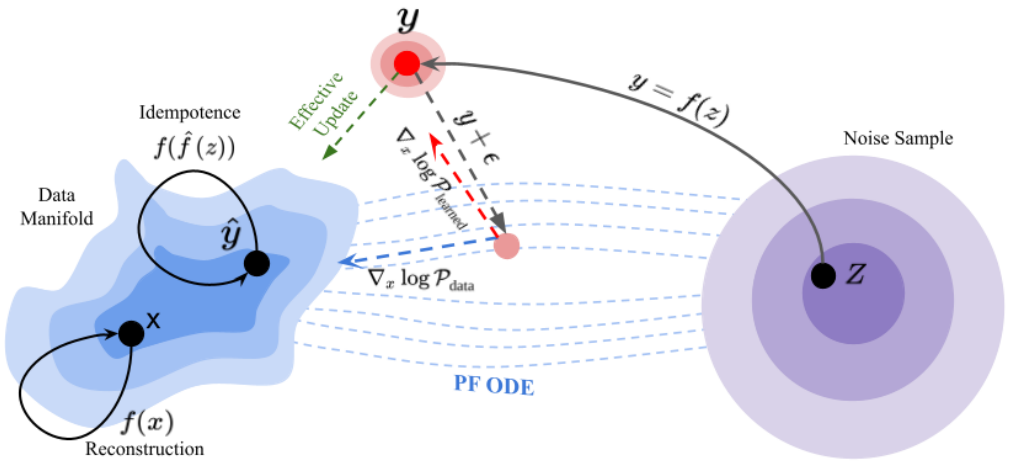


Figure 1: The goal of the optimal idempotent network, $\hat{f}$, is to project inputs outside of the target manifold on the left, $z \sim Z$, onto that manifold. While imposing dataset reconstruction $\left(\hat{f}(x) = x\right)$ and idempotence $\left(f\left(\hat{f}(z)\right) = \hat{f}(z)\right)$, our method uses estimation of the real score, $\nabla_x \log \mathcal{P}_{\text{real}}$. The projected output $y = f(Z)$ is on the data manifold when our model learned score function, $\nabla_x \log_{\text{learned}}$ is equal to $\nabla_x \log \mathcal{P}_{\text{real}}$. We design our training algorithms to estimate the real score function and train the model.

To enable exploration of IGNs on diverse domains and large-scale high-resolution datasets, we must improve the training characteristics of the optimization methods. Inspired by the success of diffusion models, we propose an optimization algorithm to stabilize IGN training. Drawing from consistency models Song et al. [2023], which learn to map points along a probability flow trajectory, we introduce Score-based Idempotent Generative Networks (SIGNs). SIGNs are trained to map noisy samples back onto the data manifold. They can be viewed as implicit time consistency models that support arbitrary noise schedules. We reformulate the IGN objective as a projection problem: noisy samples, which lie far from the data manifold (Fig. 1), are projected back onto it by the model, which remains idempotent for samples already on the manifold. This connection to score models enables the transfer of architectures, training techniques, and pre-trained weights to IGNs, while their single-step generation capabilities significantly improve sampling speed. SIGNs can be trained independently or by distilling a pre-trained diffusion model. They are capable of single-step and multi-step sampling and zero-shot editing. We establish a connection between diffusion models (or,

equivalently, score-based models) and IGNs and propose an alternative objective for training IGNs efficiently.

Contributions In our current work, we present: **(a)** a new, stable objective combining score-matching and tightening losses to replace the unbounded, unstable tightening loss in IGNs; **(b)** a theoretical analysis of our proposed objective highlighting; and **(c)** empirical validation of the new objective show our models have state of the art generation results for idempotent models and strong zero-shot editing capabilities. We achieve more than a 41% reduction in FID compared to the SOTA IGN model.

The manuscript is organized as follows: first, we describe the probability flow ordinary differential equation and idempotent models that underpin our study. We then introduce our novel learning objectives and provide theoretical insights into them. We then contextualize our contributions with a review of related work. Finally, we provide empirical experiments showcasing state-of-the-art performance for IGN models and discuss future research directions.

2 Background

Our work focuses on establishing a connection between diffusion or score-based models and IGNs to improve IGN training stability. We achieve this by training on samples along the probability flow differential equation (PF-ODE) Song et al. [2020a] defined by score-based models learning to project onto an idempotent data manifold.

Notation. We denote the unknown, true data distribution with $\mathcal{P}_{\text{data}}$, and the corresponding score of the probability density is defined as $\nabla_{\mathbf{x}} \log \mathcal{P}_{\text{data}}$. $\mathbf{x} \sim \mathcal{P}_{\text{data}}$ are objects sampled from the data distribution. The score function, $s_\phi(\cdot)$, is a parameterized function trained to approximate $\nabla_{\mathbf{x}} \log \mathcal{P}_{\text{data}}$. $O(\mathbf{x}, t) = \mathbf{x} \circledast \mathcal{N}(0, \sigma(t)\mathbf{I})$ is the noising operator, where $\sigma(t)$ is the noise schedule defined as a monotonically increasing function of time t and $\circledast$ denotes the convolution of the two distributions. $\mathcal{P}_{\text{data}}^{\sigma(t)}(\mathbf{x}_t)$ denotes the noising operator perturbed data distribution at time t . Our models are trained to learn $\mathcal{P}_{\text{model}}$, under the condition that it is sufficiently similar to $\mathcal{P}_{\text{data}}$. We use $\mathcal{U}[[1, N]]$ as the uniform distribution over the set $\{1, 2, \ldots, N\}$. The sequence of time is discretized with the set $\{t_i\}_{i=0}^{N}$, where $t_0 = \epsilon$ and $t_N = T$.

2.1 Probability Flows ODE

Following the notations from Franceschi et al. [2023a], denoising Score-based Diffusion models are represented by the following Stochastic Differential Equation (SDE):

$$dx_t = 2\sigma(t)\dot{\sigma}(t)\nabla \log \mathcal{P}_{\text{data}}^{\sigma(t)}(\mathbf{x}_t) + \sqrt{2\sigma(t)\dot{\sigma}(t)}dW_t,$$

where $\mathcal{P}_{\text{data}}^{\sigma(t)}(\mathbf{x}_t)$ denotes the Gaussian perturbed data distribution, $\sigma(t)$ is the noise schedule function that defines the noise level at time t, and the dot denotes a time derivative, and W_t denotes the Wiener Process. An important property of this SDE is that there exists a corresponding Ordinary Differential Equation, named Probability Flow ODE (PF-ODE), whose solution trajectory has the same marginal probability distribution as the SDE. This admits a PF-ODE:

$$\frac{d\mathbf{x}}{dt} = -\sigma(t)\dot{\sigma}(t)\nabla_{\mathbf{x}} \log \mathcal{P}_{\text{data}}^{\sigma(t)}(\mathbf{x}_t). \tag{1}$$

Eq. 1 enables evolving a sample from $x_{\sigma(t_a)} \sim \mathcal{P}_{\text{data}}^{\sigma(t_a)}$ to a sample $x_{\sigma(t_b)} \sim \mathcal{P}_{\text{data}}^{\sigma(t_b)}$(or equivalently noise scales $\sigma(t_a) \to \sigma(t_b)$). The goal of score-based generative methods is to flow samples from an easily sampleable distribution (like a Gaussian) to the true data distribution. Generally, the probability flows are constrained such that samples are reversed from time T, where $\mathcal{P}^{\sigma(T)} \approx \mathcal{N}(0, \sigma(T)\mathbf{I})$ to time ϵ where $\mathcal{P}^{\sigma(\epsilon)} \approx \mathcal{P}_{\text{data}}$. Flow-matching models Liu et al. [2022a] further generalize the mechanism of transporting samples from one distribution to another based on ODE transport.

In practice, we don't know the true $\mathcal{P}_{\text{data}}^{\sigma(t)}$, and thus the function, $s_\phi(\mathbf{x}_t, \sigma(t))$, parameterized by learnable weights ϕ, is trained to approximate $\nabla_{\mathbf{x}} \log \mathcal{P}_{\text{data}}^{\sigma(t)}$ to obtain the empirical PF-ODE. Several ODE solving techniques Karras et al. [2022], Lu et al. [2022], Liu et al. [2022b], Zhang et al. [2022]

have been proposed for the empirical PF-ODE. For example, using Euler's first-order method, the empirical PF-ODE updates can be written as,

$$\frac{\mathbf{x}_{t_n} - \mathbf{x}_{t_{n+1}}}{t_n - t_{n+1}} = -\sigma(t_{n+1})\dot{\sigma}(t_{n+1})s_\phi(\mathbf{x}_{t_{n+1}}, \sigma(t_{n+1})), \tag{2}$$

where the time horizon is partitioned into $N-1$ sub-intervals, and each time step is indexed by n. More details of the discretization step can be found in Karras et al. [2022]. New samples are generated by iterating randomly sampled inputs with a score model s_ϕ and an ODE solver. We can see that the trained model provides us with a strong proxy for the real score function, $\nabla_{\mathbf{x}} \log \mathcal{P}_{\text{data}}^{\sigma(t)}(\mathbf{x}_t)$.

2.2 Idempotent Generative Networks

Idempotent generative networks (IGNs) Shocher et al. [2023] are generative models based on the property of idempotence. An idempotent mapping, f, is an operator in some space $\mathcal{X}$ such that for some $x \in \mathcal{X}$, we have, $f(f(x)) = f(x)$. Identity and absolute value are canonical examples of idempotent operators. IGN learn a constrained idempotent mapping that is idempotent for elements of some target data manifold (e.g., real images), while projecting all other inputs to the manifold. The trained model can be used for generation by sampling from a distribution such as Gaussian noise and using the model to project the random sample to the learned data manifold. Specifically, the IGN optimization objectives rely on three main principles: **(a)** the identity boundary condition on the data manifold described above, **(b)** idempotence, and **(c)** the size of the data manifold is minimal. These objectives can be optimized by their respective loss functions.

Reconstruction Loss. For a sample $x \sim \mathcal{P}_{\text{data}}$, given a distance metric function $D(\cdot,\cdot)$, the reconstruction loss is:

$$\mathcal{L}_{\text{recon}} = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{P}_{\text{data}}}[\delta_\theta(\mathbf{x})] = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{P}_{\text{data}}}[D(\mathbf{x}, f_\theta(\mathbf{x}))]. \tag{3}$$

This imposes the boundary condition of preserving in-distribution samples on the data manifold.

Idempotent Loss. For any input from the domain distribution $z \sim \mathcal{Z}$, similarly with a distance metric function $D(\cdot,\cdot)$, the idempotent loss is:

$$\mathcal{L}_{\text{idem}} = \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathcal{P}_{\mathbf{z}}}[\delta_{\theta'}(f_\theta(\mathbf{z}))] = \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathcal{P}_{\mathbf{z}}}[D(f_\theta(\mathbf{z}), f'_\theta(f_\theta(\mathbf{z})))], \tag{4}$$

where f'_θ is the frozen copy of the model f_θ. $\mathbf{z}$ is restricted to be from some easily sampleable distribution such as $\mathcal{N}(\mathbf{0}, \mathbf{I})$. This loss is minimized when for any sample, z, the model is idempotent: $f_\theta(f_\theta(z)) = f_\theta(z)$.

Tightness Loss. For any input from the domain distribution $z \sim \mathcal{Z}$, the tightness loss is:

$$\mathcal{L}_{\text{tight}} = \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathcal{P}_{\mathbf{z}}}[\delta_\theta(f_{\theta'}(\mathbf{z}))] = \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathcal{P}_{\mathbf{z}}}[-D(f_\theta(f'_\theta(\mathbf{z})), f'_\theta(\mathbf{z}))]. \tag{5}$$

As above, f'_θ is the same as f_θ. The IGN training algorithm and gradient equations are reproduced in Appendix C for clarity and completeness. The IGN loss function is, therefore,

$$\mathcal{L}_{\text{IGN}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{idem}} + \lambda_t \mathcal{L}_{\text{tight}}, \tag{6}$$

where λ_t is the weight of the tightening loss term. The authors set $\lambda_t < 1$ to stabilize training.

The training of IGN requires all terms of Eq. 6 to be minimized simultaneously. This empirically leads to training instabilities. For one, the $\mathcal{L}_{tight}$ objective is the opposite of the $\mathcal{L}_{idem}$ objective, introducing adversarial optimization and making training more difficult. Furthermore, the minimum of $\mathcal{L}_{tight}$, without reweighting, is unbounded, which can lead to arbitrarily large gradient updates. Even with the reweighted tightening loss in Shocher et al. [2023] can produce large gradient updates when both $\mathcal{L}_{\text{tight}}$ and $\mathcal{L}_{\text{rec}}$ are large positive numbers, and lead to unstable training dynamics. Shocher et al. [2023] note IGNs have similar training characteristics to GANs, which are well known to suffer from training instability and mode collapse Kodali et al. [2017], Salimans et al. [2016], Arjovsky and Bottou [2017], Saxena and Cao [2021].

3 Score-based Idempotent Model

Score-based Idempotent Generative (SIGN) models distill pre-trained score models to improve IGN training dynamics and enable fast sampling. As previously mentioned, the adversarial tightening loss

is a key cause of poor training dynamics on IGNs. Crucially, we note that the IGN objective can be achieved using the score function learned by diffusion models. Specifically, for a Diffusion or flow-matching model, we have a solution trajectory $\{\mathbf{x}_t\}_{t\in[\epsilon,T]}$ based on the corresponding PF-ODE as in Eq.1. On this trajectory, only the point x_ϵ is on the data distribution $\mathcal{P}_{\text{data}}$, while the rest of the points on the trajectory are off the manifold and naturally form a constraint on the size of the data manifold.

The main contribution of our work is the **score-based idempotent loss**, $\mathcal{L}_{\text{SIGN}}$, for training idempotent generative models. We replace the unbounded tightening loss with a **distribution matching loss**, $\mathcal{L}_{\text{dmd}}$, or a **consistency flow loss**, $\mathcal{L}_{\text{flow}}$, to learn the data manifold from a score-function estimate for an idempotent generative model.

Distribution Matching Loss. The goal of a generative model is to ultimately sample from the data distribution by sufficiently emulating some target distribution. A diffusion or flow-based learns a target distribution $\mathcal{P}_{\text{target}}$ that is sufficiently close to $\mathcal{P}_{\text{data}}$. While we do not have access to either distribution, we have access to a learned approximation of $\nabla_{\mathbf{x}_t}\log\mathcal{P}^{\sigma(t)}_{\text{data}}(\mathbf{x}_t)$ through the trained diffusion model. By matching the scores of our idempotent model over a family of noisy distributions with the scores of our pre-trained model,

Following Yin et al. [2024a], we also estimate the probability densities by doing gradient updates using approximate scores. We directly

$$\nabla_\theta\mathcal{L}_{\text{DMD}} = \mathop{\mathbb{E}}_{\substack{\mathbf{z}\sim\mathcal{N}(0,\mathbf{I}),\\ n\sim\mathcal{U}[[1,N]]}} \left(s_{\text{learned}}(y_n,n) - s_{\text{diffusion}}(y_n,n)\right)\frac{df_\theta}{d\theta} \tag{7}$$

where, $\mathbf{y}_n = O(f_\theta(\mathbf{z}),t_n.)$ $O(\mathbf{x},t)$ is the noising operator, such that, $O(\mathbf{x},t) = \mathbf{x}\circledast\mathcal{N}(0,\sigma(t)\mathbf{I})$, which performs t steps forward diffusion on the given input $\mathbf{x}$. We use a pre-trained diffusion model for $s_{\text{diffusion}}(,)$. An auxiliary diffusion model is trained along with the SIGN model to provide the learned score estimates, $s_{\text{learned}}(,)$.

Consistency Flow Loss. $\mathcal{L}_{\text{DMD}}$ requires an additional model tracking the scores of the current model during training, incurring a large computation cost. As an alternative, we take inspiration from consistency models to minimize the size of the manifold through the probability flow ODE and propose the flow loss. Based on a diffusion-based ODE solver to impose restrictions on the learned manifold, the flow-based loss is given by:

$$\mathcal{L}_{\text{Flow}} = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{P}_{\text{data}},n\sim\mathcal{U}[[1,N]]}[D(f_\theta(\mathbf{x}_{t_n}),f_{\theta'}(\mathbf{x}_{t_s}))] \tag{8}$$

where, $\mathbf{x}_{t_n} = O(\mathbf{x},t_n)$ and $\mathbf{x}_{t_s}$ is obtained by taking a step following the empirical PF-ODE in Eq. 2 using a learned score function, like a trained diffusion model, or an empirical score-estimator. The real score function, $\nabla_x\log\mathcal{P}_{\text{data}}$, defines a vector field over the data space x. By taking the expectation over $\mathbf{x}$ and noises t_n, we obtain the average direction of motion at any $\mathbf{x}$, which is used as the score function estimate. Given constraints, such as linearity in the trajectory, as shown in Theorem 3, this allows us to learn the target manifold. Intuitively, this loss can be seen as treating points along the trajectory as off-manifold domains, and we require them to be mapped in one step onto the same point of the data manifold. We use the training techniques from Consistency Models, such as using pre-trained diffusion models, or an unbiased estimator for the ODE-solvee to approximate the score function.

Improving Training Dynamic While the aforementioned objectives are sufficient in theory, we introduce further improvements to speed up convergence and improve generation quality. We use 2 auxiliary loss terms **(a)** regression loss and **(b)** denoising loss for this purpose. Yin et al. [2024a] show that the regression loss of supervised learning on generated pairs from a pre-trained model significantly improves model quality. Similarly, the denoising loss connects the training objective to a test-time scenario where noisy samples may be input to the model to iteratively improve generated outputs.

Score-based Idempotent Loss. Using the consistency loss over the modified distributions obtained by adding Gaussian noise to the data distribution, we propose the consistency-based IGN loss as:

$$\mathcal{L}_{\text{SIGN}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{idem}} + \lambda_f\mathcal{L}_{\text{flow}} + \lambda_d\mathcal{L}_{\text{dmd}} + \lambda_r\mathcal{L}_{\text{reg}} + \lambda_n\mathcal{L}_{\text{denoise}} \tag{9}$$

where, $\lambda_f, \lambda_d, \lambda_r, \lambda_n$ are hyperparameters for each auxiliary loss terms. In practice, we set $\lambda_f, \lambda_d, \lambda_r, \lambda_n \in [0, 1]$, to optionally enable various loss terms. Depending on the training environment and dataset complexity, a subset of the loss terms is used. The coefficients are decided heuristically so that all terms have the same magnitude at the start of training.

We replace the unbounded tightening loss in Eq. 5 with a combination of distribution matching and flow-based losses in Eq.9. The tightening loss in the original IGN aims to restrict the manifold to only include the data distribution, but causes training instability. We directly restrict the learned data manifold toward the data distribution. By replacing the unbounded loss and directly restricting the learned data manifold, we ensure all components of our objective are bounded for stable training. In the following section, we show in Theorem 1 and Theorem 2, the distribution matching and flow-based losses accomplish the same objective. The training pseudo-code for the complete training loss and all constituent losses is in Appendix A.1.

Sampling SIGN aims to excel at single-step generation, which is its primary mode of operation. However, similar to Consistency models, IGN, and Diffusion models, SIGN also provides the ability to perform multi-step sampling to trade off computational cost for generation quality. Generated outputs can be iteratively refined by computing multiple forward passes on the data. The recursive sampling algorithm is presented in Alg. 2.

Karras et al. [2022] show that additional noise injected during the sampling process improves the quality of the generated images. As the signal-to-noise ratio of the initial sample is 0, the additional noise during sampling allows the model to correct imperfections during the early stages of generation. In algorithm 3, we provide an additional sampling method that effectively "pushes" the sample out of the manifold. The additional noise injection cancels out the error introduced when sampling from a low SNR region. Different use cases may be beneficial for different sampling approaches. With inputs with a high signal-to-noise ratio, such as the case for partially corrupted or low-resolution images, the recursive sampling approach may be more favorable. On the other hand, unconditional generation may benefit from backtracking and adding additional noise in the sampling procedure in Algorithm 3.

4 Theoretical Analysis

In this section, we provide convergence guarantees and error bounds and build on our understanding of the proposed SIGN model.

Theorem 1. *Given a trained SIGN model, f_θ, such that it is a measurable idempotent map, $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$. Let $\mathcal{P}_{data}$ be the true data distribution and $\mathcal{P}_{f_\theta} := f_\theta \# \mathcal{P}_{data}$ its pushforward through f_θ. Given the regular and k-dimensional connected, $\mathcal{C}^2$ manifolds D and M, we have $\operatorname{supp} \mathcal{P}_{data} = D$ and $\operatorname{supp} \mathcal{P}_f = M$, and the score functions are defined on $\forall x \in D \cap M$, if θ is the global minimum $\mathcal{L}_{SIGN}$, then $D = M$.*

The proof consists of showing that the support of the learned distribution and data distribution are included in each other the densities are equal on the manifold. The full proof is included in Appendix B. In practice, a pre-trained model or an empirical score estimator is used to obtain the data score function. Score models may not cover the whole manifold, resulting in bias. Intuitively, for points not on the data manifold, the real score function has non-zero gradients, pushing the pushforward distribution towards the real distribution, contracting the manifold. Interestingly, Kamb and Ganguli [2024], Biroli et al. [2024], and De Bortoli [2022] analytically show that diffusion models recover target distributions on low-dimensional manifolds. Empirically, our method attempts to learn a mapping to this learned manifold.

Next, we look at the convergence conditions for our proposed flow-based SIGN loss.

Theorem 2. *Denote the distribution learned by the trained SIGN model f_θ as $\mathcal{P}_\theta$. Assuming a large enough model capacity such that:*

$$\exists \theta^* = \arg\min_\theta \mathcal{L}_{recon} = \arg\min_\theta \mathcal{L}_{flow} = 0$$

then the learned distribution $\mathcal{P}_\theta = \mathcal{P}_{data}$, the true data distribution.

We can see from Theorem 2 that imposing consistency across noise levels of the PF-ODE trajectories can sufficiently tighten the manifold to capture the underlying data distribution. But we must note

that the above holds for a case of sufficiently large models, perfect projection at all noise levels, non-overlapping trajectories, as well as an infinitely accurate discretization of the PF-ODE. Unfortunately, in practice, such conditions are not feasible. Particularly, obtaining trajectories by reversing the pre-trained score model is prohibitively expensive. This necessitates adding additional loss terms as described in the previous section to improve training dynamics. The quality of the estimate thus decides the applicability of the learned SIGN. In the next theorem, we show that if the learned score function can generate samples of empirical PF-ODE trajectories with errors uniformly bounded by some noise-related quantity, we can guarantee that the error of the learned SIGN to the optimal idempotent function is bounded.

Theorem 3. *Let $\Delta = \max |\sigma(t_{n+1}) - \sigma(t_n)|$ for $n \in \{0, N-1\}$ and f be the optimal idempotent function. For some learned model f_θ which satisfies the L-Lipschitz condition. Denote $\{\mathbf{x}_t\}_{t\in[\epsilon,T]}$ the exact PF-ODE trajectory by updating using Eq. 1, and $\{\hat{\mathbf{x}}_t\}_{t\in[\epsilon,T]}$ the empirical results by Eq. 2 (i.e., using $\mathbf{x}_{t_{n+1}}$ to solve step n in Eq.2 gives the resulting $\hat{\mathbf{x}}_{t_n}$). Assume the local approximation error of updating PF-ODE, $||\hat{\mathbf{x}}_{t_n} - \mathbf{x}_{t_n}||_2$, is uniformly bounded by, $\mathcal{O}((\sigma(t_{n+1}) - \sigma(t_n))^{p+1})\, \forall n \in \{1, N-1\}$ with $p \geq 1$, and $L \in \mathbb{R}_{\geq 0}$, . If $\mathcal{L}_{Flow}(\theta) = 0$, and $\mathcal{L}_{Recon}(\theta) = 0$, then we have,*

$$\sup_{\mathbf{x}_{t_n}} ||f_\theta(\mathbf{x}_{t_n}) - f(\mathbf{x}_{t_n})||_2 = \mathcal{O}((\Delta)^p)$$

Similar to Song et al. [2023], we base the proof on the global error bounds for numerical ODE solvers. Due to space limitations, we present the full proofs in Appendix B. Intuitively, Theorem 3 shows an important characteristic of SIGNs and ODE trajectories. As the upper-bound error is dependent on the truncation error of the trajectories, paths with high curvature will have high error. Therefore, while the SIGN algorithm works for all diffusion, score, and flow models, algorithms with linear trajectories are better suited. This falls in line with observations in Karras et al. [2022], Liu et al. [2022a], Liu [2022], Lee et al. [2024a].

5 Related Work

IGNs, GANs, and Stability. Our work focused on improving the training of IGN, a model class for fast, single-step generation. IGN assumes an underlying data manifold and proposes a transformation f that maps any input source to the manifold. In addition, all the data points on the manifold have to be mapped to itself (thus, idempotent) while minimizing the size of the manifold. Jensen and Vicary [2025] present a modified backpropagation method to enforce idempotency on generative networks based on perturbation theory for MLP and CNN-based networks, but still suffer from the sensitivity to training dynamics. IGNs are similar GANs as both contain adversarial loss terms. IGNs are particularly similar to EB-GAN Zhao et al. [2016], but IGN doesn't require the input source to be a random noise. The instability of GAN-like models is well documented in literature [Durall et al., 2020, Yu et al., 2020, Zhao et al., 2018], and significant care is required to train large-scale adversarial models. Interestingly, Franceschi et al. [2023b] show a similar connection between GAN models and score models, where they train GANs using pre-trained models. We provide improved performance, additional theoretical guarantees for IGNs using score models, as well as practical training recipes.

Diffusion, Score-based Models, and Distillation. Diffusion models Sohl-Dickstein et al. [2015], Song et al. [2020a] produce high-quality images through a slow and iterative process, which incurs high computational costs. To mitigate this challenge, fast sampling and model distillation methods are of great interest. Consistency Models Song et al. [2023] enable single-step generation by mapping all points on a PF-ODE trajectory to a single output. We show the connection between Consistency Models and IGNs and propose a novel loss to stabilize the training of the IGN models. Furthermore, Lipman et al. [2022], Liu et al. [2022c], Lee et al. [2024b] flow matching similarly casts generative modeling as a transport mapping problem, where straight trajectories in rectified flows improve sampling efficiency. Idempotent models can similarly be distilled from flow-matching models as well.

6 Results

We train multiple SIGN models on MNIST, CIFAR-10, and CelebA datasets to get empirical validation for our theoretical results. We also use pre-trained models to perform zero-shot editing. The training details are described in Appendix D.

(a) MNIST (Diffusion) (b) MNIST (Estimator) (c) CIFAR-10 (single-step) (d) CelebA (single-step)

Figure 2: Results of unconditional sampling from SIGN models. (a) Single and Multi-step generation of a SIGN model trained on MNIST using a diffusion model as the score function. (b) Single and Multi-step generation using an empirical score estimator as the score function. (c) Random sampling of single-step generation from a SIGN model trained on CIFAR-10. (d) Random sampling of single-step generation from a SIGN model on CelebA.

Image Generation. We evaluated our model's image generation capabilities on three standard benchmarks: MNIST Deng [2012], CIFAR-10 Krizhevsky et al. [2009], and CelebA Liu et al. [2015]. These datasets consist of 28x28 grayscale images, 32x32 color images, and 64x64 color images, respectively.

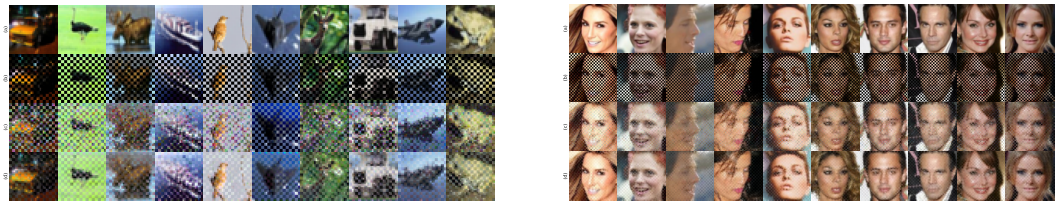
For MNIST, we used a convolutional U-Net as the model with and without time embeddings for the pre-trained diffusion model and the distilled SIGN model, respectively. We used a ℓ_2 distance as our distance metric. We use a pre-trained diffusion model and an unbiased score estimator to train our SIGN models for MNIST. For the simple dataset, we do not use distribution matching, regression, or denoising loss. As illustrated in Fig. 2a and Fig. 2b, our SIGN models are capable of unconditional generation in a single pass, and image quality is progressively enhanced with multiple passes.

For CIFAR-10 and CelebA, we adapted model structures from EDM Karras et al. [2022] and DDIM Song et al. [2022] to make use of their pre-trained weights as a starting point for our training. We show the result of single-step unconditional generation in Fig. 2c and Fig. 2d. Following Alg. 3 and 2, we iteratively sample to generate samples. As the samples are more challenging, we use an additional regression loss term to stabilize the training. To evaluate performance, we first generated 50,000 unconditional, single-step samples from our trained model for each dataset, then we calculated the Fréchet Inception Distance (FID) using 2048 features to measure the similarity of the generated samples to the target dataset. Our model sets a new state-of-the-art benchmark for idempotent models by achieving FID scores of 11.09 on CIFAR-10 and 23.32 on CelebA. Our CelebA FID score significantly outperforms the original IGN model (FID=39). Prior work on idempotent models in Shocher et al. [2023] and Jensen and Vicary [2025] do not train on CIFAR-10; as such, we are the first to report CFIAR-10 results for idempotent models. For our initial ablation study, we trained a model on the CelebA dataset for the same 350 epochs but without our proposed loss. The model achieved an FID score of 123.70, which indicates that our proposed loss significantly improves the training dynamics.

Zero-shot Editing. We investigated the zero-shot image editing capabilities of our SIGN models on the CelebA and CIFAR-10 datasets. As shown in Fig. 3b and Fig. 3a, we first applied a checkerboard binary mask to corrupt the data, and tested how the model would perform in single-step and multi-step scenarios. For multi-step sampling, we applied a customized 10-step noise schedule. The model was able to project the corrupted image back towards the target data manifold from single-step sampling, despite not being specifically trained for this task. The multi-step results further improve the image quality, yielding a result closer to the original. We should point out that because CelebA has a higher resolution than CIFAR-10, the defects would be less obvious in Fig. 3b than in Fig. 3a.

7 Limitations

Though not competitive with general SOTA generative models, we perform better than current IGN models. We show that IGN models can be stably trained using non-adverserial losses. As Jensen and Vicary [2025] show, idempotent networks are a nascent line of generative models, and there is significant room for improvement in the inductive biases and training methods required to optimize



(a) CIFAR-10. (b) CelebA

Figure 3: Zero-shot masked image editing with a SIGN model trained. From *top* to *bottom*: (1) Original images. (2) Masked inputs. (3) Single-step sampling results. (4) Multi-step sampling results.

these models. Intuitively, using a single network to learn both a large projection mapping from noise to the data manifold, while also learning identity mapping on the manifold, is difficult, as the objective is significantly different from the usual generative modeling tasks. As a result, we cannot take advantage of the standard practices of generative models and the wealth of accumulated knowledge from the usual generative modeling community. We hypothesize, transformer-based and especially mixture-of-experts-based Chen et al., 2022, Shazeer et al., 2017, Riquelme et al., 2021 networks may provide significantly improved performance and bring IGNs on par with other single-step and distilled models. We plan on exploring architectural choices to improve model capabilities along with larger models, higher training compute budgets, and larger, modern datasets. Furthermore, our work greatly improves the training stability of idempotent generative models, which is a key requirement in enabling future work on high-resolution datasets. The regression loss requires pre-generation of a large set of images, requiring a large amount of compute and memory. However, the regression loss is not required, as Yin et al. [2024b] shows that regression loss is not required to obtain strong generation results. We also acknowledge that our initial ablation study is limited due to computational resource constraints, but we plan to further investigate it when resources permit.

8 Future Work

A key feature of our work is the improved training dynamics of idempotent generative models by utilizing learned score functions. Due to training instability in the adversarial loss, prior work on IGNs is difficult to reproduce and focused on simpler datasets. Optimizing modeling choices and training method is challenging when training is unstable. We hope our work will enable large-scale architectural optimization studies for idempotent networks that may be transferred to conventional IGN training as well. Furthermore, the inductive bias of idempotent networks is a natural fit for many other learning and generative tasks, especially in scientific workloads.

SIGN models can be combined with existing score-based models in two ways: either by "fast-forwarding" the reverse process by inputting partially denoised samples to the SIGN generator, similar to denoising diffusion GANs Xiao et al. [2021], or by employing multi-step iterative sampling schemes inspired by Shocher et al. [2023] and Song et al. [2023]. As noted in Shocher et al. [2023], the model may also benefit from a two-step approach as in Rombach et al. [2022] instead of directly applying to the pixel-space. Furthermore, flow-matching methods such as rectified flows Liu et al. [2022c] provide an attractive alternative to diffusion models as teacher models due to their straight trajectories.

9 Conclusion

In this work, we connect Idempotent generative models with score-based diffusion models. Our proposed new losses to train IGN models improve the training characteristics of IGNs and provide theoretical guarantees of our optimization methods, and strong empirical results for idempotent models. We provide a first baseline on CIFAR-10, as well as improving SOTA CelebA FID by more than 40% for idempotent models. We view this work as an initial step towards connecting IGNs with score-based generative models that allow the development of more powerful models. Connecting to score-based opens the door for transferring learning techniques, model architecture, and even learned weights to improve IGN models. A more stable training algorithm can enable further exploration

into identifying network architectures that are suitable for idempotent models and allow learning on large-scale, higher-resolution datasets. We plan on exploring such possibilities in the future.

References

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.

Ning Yu, Ke Li, Peng Zhou, Jitendra Malik, Larry Davis, and Mario Fritz. Inclusive gan: Improving data and minority coverage in generative models. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 377–393. Springer, 2020.

Shengjia Zhao, Hongyu Ren, Arianna Yuan, Jiaming Song, Noah Goodman, and Stefano Ermon. Bias and generalization in deep generative models: An empirical study. *Advances in Neural Information Processing Systems*, 31, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*, pages 2722–2730. PMLR, 2019.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

Younghyun Jo, Sejong Yang, and Seon Joo Kim. Investigating loss functions for extreme super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 424–425, 2020.

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022a.

Assaf Shocher, Amil Dravid, Yossi Gandelsman, Inbar Mosseri, Michael Rubinstein, and Alexei A Efros. Idempotent generative network. *arXiv preprint arXiv:2311.01462*, 2023.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022b.

Kangfu Mei and Vishal Patel. Vidm: Video implicit diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9117–9125, 2023.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

Arne Schneuing, Yuanqi Du, Charles Harris, Arian Jamasb, Ilia Igashov, Weitao Du, Tom Blundell, Pietro Lió, Carla Gomes, Max Welling, et al. Structure-based drug design with equivariant diffusion models. *arXiv preprint arXiv:2210.13695*, 2022.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pages 8867–8887. PMLR, 2022.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models, 2023. URL `https://arxiv.org/abs/2303.01469`.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020a.

Jean-Yves Franceschi, Mike Gartrell, Ludovic Dos Santos, Thibaut Issenhuth, Emmanuel de Bézenac, Mickaël Chen, and Alain Rakotomamonjy. Unifying gans and score-based diffusion as generative particle models. *arXiv preprint arXiv:2305.16150*, 2023a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022a.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022b.

Qinsheng Zhang, Molei Tao, and Yongxin Chen. gddim: Generalized denoising diffusion implicit models. *arXiv preprint arXiv:2206.05564*, 2022.

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

Divya Saxena and Jiannong Cao. Generative adversarial networks (gans) challenges, solutions, and future directions. *ACM Computing Surveys (CSUR)*, 54(3):1–42, 2021.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6613–6623, 2024a.

Mason Kamb and Surya Ganguli. An analytic theory of creativity in convolutional diffusion models. *arXiv preprint arXiv:2412.20292*, 2024.

Giulio Biroli, Tony Bonnaire, Valentin De Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1):9957, 2024.

Valentin De Bortoli. Convergence of denoising diffusion models under the manifold hypothesis. *arXiv preprint arXiv:2208.05314*, 2022.

Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

Sangyun Lee, Zinan Lin, and Giulia Fanti. Improving the training of rectified flows. *Advances in neural information processing systems*, 37:63082–63109, 2024a.

Nikolaj Banke Jensen and Jamie Vicary. Enforcing idempotency in neural networks. In *Forty-second International Conference on Machine Learning*, 2025.

Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

Ricard Durall, Avraam Chatzimichailidis, Peter Labus, and Janis Keuper. Combating mode collapse in gan training: An empirical analysis using hessian eigenvalues. *arXiv preprint arXiv:2012.09673*, 2020.

Jean-Yves Franceschi, Mike Gartrell, Ludovic Dos Santos, Thibaut Issenhuth, Emmanuel de Bézenac, Mickaël Chen, and Alain Rakotomamonjy. Unifying gans and score-based diffusion as generative particle models. *arXiv preprint arXiv:2305.16150*, 2023b.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022c.

Sangyun Lee, Zinan Lin, and Giulia Fanti. Improving the training of rectified flows. *Advances in neural information processing systems*, 37:63082–63109, 2024b.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL `https://arxiv.org/abs/2010.02502`.

Zixiang Chen, Yihe Deng, Yue Wu, Quanquan Gu, and Yuanzhi Li. Towards understanding the mixture-of-experts layer in deep learning. *Advances in neural information processing systems*, 35: 23049–23062, 2022.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024b.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020b.

A Algorithm

A.1 Training Algorithms

Algorithm 1 Consistent Idempotent Training

Input: Dataset $\mathcal{D}$, models f_θ, $f_{\theta'}$, distance metric, $D(\cdot,\cdot)$, noising operator $O(\cdot,\cdot)$, learning rate η, loss term hyperparameters $\lambda_f, \lambda_d, \lambda_r, \lambda_n$
while not converged **do**
  Copy $f_{\theta'} \leftarrow f_\theta$
  Sample $\mathbf{x} \sim \mathcal{D}$
  Sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Sample $n \sim \mathcal{U}([[1, T]])$
  $\mathbf{x}_{t_n} \leftarrow O(\mathbf{x}, t_n)$
  Obtain $\mathbf{x}_{t_s}$ from solving steps in Eq. 2.
  $\mathbf{x}_{\text{recon}}, \mathbf{x}_{\text{sample}} \leftarrow f_\theta(\mathbf{x}), f_\theta(\mathbf{z})$
  $\mathbf{x}_{\text{idem}} \leftarrow f_\theta(f_{\theta'}(\mathbf{z}))$
  Copy $\mathbf{x}_{\text{clone}} \leftarrow x_{\text{sample}}$
  $y_n \leftarrow O(f_\theta(\mathbf{x}_{\text{sample}}), t_n.)$
  $\mathcal{L}_{\text{recon}} \leftarrow D(\mathbf{x}, \mathbf{x}_{\text{recon}})$
  $\mathcal{L}_{\text{idem}} \leftarrow D(\mathbf{x}_{\text{sample}}, \mathbf{x}_{\text{idem}})$
  $\mathcal{L}_{\text{denoise}} \leftarrow D(\mathbf{x}, f_\theta(\mathbf{x}_{t_n}))$
  $\mathcal{L}_{\text{flow}} \leftarrow D(f_\theta(\mathbf{x}_{t_n}), f_{\theta'}(\mathbf{x}_{t_s}))$
  $\mathcal{L}_{\text{DMD}} \leftarrow D(s_{\text{learned}}(y_n, n) - s_{\text{diffusion}}(y_n, n))$
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{idem}} + \lambda_f \mathcal{L}_{\text{flow}} + \lambda_d \mathcal{L}_{\text{dmd}} + \lambda_r \mathcal{L}_{\text{reg}} + \lambda_n \mathcal{L}_{\text{denoise}}$
  $f_\theta \leftarrow f_\theta - \eta \nabla_\theta \mathcal{L}(f_\theta)$
end while

Algorithm 2 Recursive Sampling

Input: Trained CIGN $\mathbf{f}_\theta(\cdot)$, initial noise $\mathbf{x}_T$
$\mathbf{x} \leftarrow \mathbf{f}_\theta(\mathbf{x})$
while not converged **do**
  $\mathbf{x} \leftarrow \mathbf{f}_\theta(\mathbf{x})$
end while

Algorithm 3 Multistep Sampling with Editing

Input: Trained CIGN $\mathbf{f}_\theta(\cdot)$, initial noised data $\mathbf{x}'$, image mask $\mathbf{M}$
Noise schedule $0 < \sigma_N < \sigma_{N-1} \ldots < \sigma_1$
$\mathbf{x} \leftarrow \mathbf{f}_\theta(\mathbf{x}') \odot M + \mathbf{x}' \odot (1 - M)$
for $i = 1, i < N$ **do**
  Sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_\tau \leftarrow \mathbf{x} + \sigma_i z$
  $\mathbf{x} \leftarrow \mathbf{f}_\theta(\mathbf{x}_\tau) \odot M + \mathbf{x} \odot (1 - M)$
end for

B Proofs

Theorem. *Given a trained SIGN model, f_θ, such that it is a measurable idempotent map, $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$. Let $\mathcal{P}_{data}$ be the true data distribution and $\mathcal{P}_{f_\theta} := f_\theta \# \mathcal{P}_{data}$ its pushforward through f_θ. Given the regular and connected manifolds D and M, we have $\operatorname{supp} \mathcal{P}_{data} = D$ and $\operatorname{supp} \mathcal{P}_f = M$, and the score functions are defined on $\forall x \in D \cap M$, if θ is the global minimum $\mathcal{L}_{SIGN}$, then $D = M$.*

Proof. We start with our trained idempotent model f_θ and the definition of the manifold M. As the θ is the global minimizer of $\mathcal{L}_{\text{SIGN}}$, we have a perfectly idempotent model. We have $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$, and

$$M := \text{Im}(f) = \{y : \exists x, y = f_\theta(x)\}$$

where every $y \in M$ is a fixed point such that $f_\theta(y) = y$. We have a learned distribution distribution $p_{f_\theta} := f_\theta \# \mathcal{P}_{\text{data}}$. We also have $\text{supp}\, \mathcal{P}_{\text{data}} = D$ and $\text{supp}\, \mathcal{P}_f = M$. Finally, since we have that θ is the global minimizer of $\mathcal{L}_{\text{SIGN}}$, crucially $\mathcal{L}_{\text{DMD}}$ is minimized. Minimizing the distribution score matching loss results in a score equality over the support of distributions.

On each C^2 manifold, with volume measures μ_M and μ_D, let's define positive densities, q_M and q_D with respect to the volume measures, μ_M and μ_D.

Therefore, the tangential scores of each manifold-density are equal:

$$\nabla_T \log q_M = \nabla_T \log q_D \quad \forall x \in D \cap M$$

Since the densities must be normalized, the normalization constant is 0, and they must be equal. Assume for a contradiction that there exists $x_o \in D$ and $x_0 \notin M$. Thus, there is an open neighborhood U of x_0 where $\mathcal{P}_{\text{data}}$ has a positive manifold density $q_D > 0$. But since $x_0 \notin M$, $q_M \leq 0$ or does not exist in U. This contradicts the equality of manifold densities; thus, there is no x_0 such that $x_0 \in D$ but $x_0 \notin M$. Thus, we have $\text{supp}(\mathcal{P}_{\text{data}}) \subseteq \text{supp}(\mathcal{P}_{f_\theta})$ and equivalently, $D \subseteq M$.

Minimizing $\mathcal{L}_{\text{SIGN}}$ also requires minimizing the $\mathcal{L}_{idem}$. As $\mathcal{P}_{f_\theta}$ is the idempotent pushforward of $\mathcal{P}_{\text{data}}$, we have $\text{supp}\, \mathcal{P}_{f_\theta} \subseteq f_\theta(\text{supp}\, \mathcal{P}_{\text{data}})$. Since, $\text{supp}\, \mathcal{P}_{\text{data}} \subseteq \text{supp}\, \mathcal{P}_{f_\theta}$, we have, $f_\theta(\text{supp}\, \mathcal{P}_{\text{data}}) \subseteq f_\theta(\text{supp}\, \mathcal{P}_{f_\theta})$. From idempotent symmetry over the support, and have $f_\theta(\text{supp}\, \mathcal{P}_{f_\theta}) = \text{supp}\, \mathcal{P}_{f_\theta}$ and we have $\text{supp}\, \mathcal{P}_{f_\theta} \subseteq \text{supp}\, \mathcal{P}_{\text{data}}$. Thus, we have $\text{supp}(\mathcal{P}_{f_\theta}) \subseteq \text{supp}(\mathcal{P}_{\text{data}})$ and equivalently, $M \subseteq D$.

Combining, $M \subseteq D$ and $D \subseteq M$, we van see that $D = M$ and $\mathcal{P}_{f_\theta} = \mathcal{P}_{\text{data}}$, completing the proof. □

Theorem. *Denote the distribution learned by the trained SIGN model f_θ as $\mathcal{P}_\theta$. Assuming a large enough model capacity such that:*

$$\exists \theta^* = \arg\min_\theta \mathcal{L}_{recon} = \arg\min_\theta \mathcal{L}_{flow} = 0$$

then the learned distribution $\mathcal{P}_\theta = \mathcal{P}_{data}$, the true data distribution.

Proof. Assuming the set of parameters θ^*, the model f_{θ^*} minimizes the proposed flow loss, in Eq. (8), and thus we have,

$$d(f_{\theta^*}(\mathbf{x}_{t_{n+1}}), f_{\theta^*}(\mathbf{x}_{t_n})) = 0,$$

where $n \in [\epsilon, T-1]$ denotes trajectory along the PF-ODE with different noise steps. By the definition of a metric function, we have,

$$f_{\theta^*}(\mathbf{x}_{t_{n+1}}) = f_{\theta^*}(\mathbf{x}_{t_n}). \tag{10}$$

Now, let's consider the base case, $n = 0$ and $t_0 = \epsilon$. We have:

$$\begin{aligned} d(f_{\theta^*}(\mathbf{x}_{t_1}), f_{\theta^*}(\mathbf{x}_\epsilon)) &= 0, \\ d(f_{\theta^*}(\mathbf{x}_{t_1}), \mathbf{x}_\epsilon) &\overset{(a)}{=} 0, \\ f_{\theta^*}(\mathbf{x}_{t_1}) &\overset{(b)}{=} \mathbf{x}_\epsilon \end{aligned} \tag{11}$$

where (a) is due to f_{θ^*} minimizing the reconstruction loss, therefore, $\forall \mathbf{x}_\epsilon$, $f_{\theta'}(\mathbf{x}_\epsilon) = \mathbf{x}_\epsilon$ and (b) is due to the definition of the distance metric. By Eq. (10), Eq. (11), and mathematical induction, we will have $f_{\theta^*}(\mathbf{x}_T) = \mathbf{x}_\epsilon$. In other words, for all random noise sampled from the source $\mathbf{x}_T \sim \mathcal{Z}$, after applying the learned CIGN transformation f_{θ^*}, will fall in the terminal distribution $\mathbf{x}_\epsilon \sim \mathcal{P}^\epsilon$, which is the data distribution, $\mathcal{P}_{\text{data}}$. □

Theorem. *Let $\Delta = \max |\sigma(t_{n+1}) - \sigma(t_n)|$ for $n \in \{0, N-1\}$ and f be the optimal idempotent function. For some learned model f_θ which satisfies the L-Lipschitz condition. Denote $\{\mathbf{x}_t\}_{t\in[\epsilon,T]}$ the exact PF-ODE trajectory by updating using Eq. 1, and $\{\hat{\mathbf{x}}_t\}_{t\in[\epsilon,T]}$ the empirical results by Eq. 2 (i.e., using $\mathbf{x}_{t_{n+1}}$ to solve step n in Eq.2 gives the resulting $\hat{\mathbf{x}}_{t_n}$). Assume the local approximation error of updating PF-ODE, $||\hat{\mathbf{x}}_{t_n} - \mathbf{x}_{t_n}||_2$, is uniformly bounded by, $\mathcal{O}((\sigma(t_{n+1}) - \sigma(t_n))^{p+1})\, \forall n \in \{1, N-1\}$ with $p \geq 1$, and $L \in \mathbb{R}_{\geq 0}$, . If $\mathcal{L}_{Flow}(\theta) = 0$, and $\mathcal{L}_{Recon}(\theta) = 0$, then we have,*

$$\sup_{\mathbf{x}_{t_n}} ||f_\theta(\mathbf{x}_{t_n}) - f(\mathbf{x}_{t_n})||_2 = \mathcal{O}((\Delta)^p)$$

Proof. Recall the $\mathcal{L}_{\text{Flow}}$:

$$\mathcal{L}_{\text{Flow}} = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{P}_{\text{data}}, n\sim\mathcal{U}[[1,N]]}[D(f_\theta(\mathbf{x}_{t_n}), f_{\theta'}(\mathbf{x}_{t_s}))].$$

$\mathcal{L}_{\text{Flow}} = 0$ implies $D(f_\theta(\mathbf{x}_{t_{n+1}}), f_\theta(\hat{\mathbf{x}}_{t_n})) = 0$ and thus $f_\theta(\mathbf{x}_{t_{n+1}}) = f_\theta(\hat{\mathbf{x}}_{t_n})$. Since f is the optimal CIGN solution, we have $f(\mathbf{x}_{t_{n+1}}) = f(\mathbf{x}_{t_n})$. Denote error at noise level n as $\mathbf{e}_n = f_\theta(\mathbf{x}_{t_n}) - f(\mathbf{x}_{t_n})$.

We now form the recursion relation,

$$\begin{aligned}
\mathbf{e}_{n+1} &= f_\theta(\mathbf{x}_{t_{n+1}}) - f(\mathbf{x}_{t_{n+1}}) && (12)\\
&= f_\theta(\hat{\mathbf{x}}_{t_n}) - f(\mathbf{x}_{t_n}) && (13)\\
&= f_\theta(\hat{\mathbf{x}}_{t_n}) - f_\theta(\mathbf{x}_{t_n}) + f_\theta(\mathbf{x}_{t_n}) - f(\mathbf{x}_{t_n}) && (14)\\
&= f_\theta(\hat{\mathbf{x}}_{t_n}) - f_\theta(\mathbf{x}_{t_n}) + \mathbf{e}_n. && (15)
\end{aligned}$$

Due to the Lipschitz condition, we have

$$||f_\theta(\hat{\mathbf{x}}_{t_n}) - f_\theta(\mathbf{x}_{t_n})||_2 \leq L||\hat{\mathbf{x}}_{t_n} - \mathbf{x}_{t_n}||_2.$$

Thus, we can bound the error at noise-scale $n+1$ with,

$$||\mathbf{e}_{n+1}||_2 \leq ||\mathbf{e}_n||_2 + L||\hat{\mathbf{x}}_{t_n} - \mathbf{x}_{t_n}||_2$$

Furthermore, as the local approximation error, $||\hat{\mathbf{x}}_{t_n} - \mathbf{x}_{t_n}||_2$, is uniformly bounded by, $\mathcal{O}((\sigma(t_{n+1}) - \sigma(t_n))^{p+1})$, and $L \in \mathbb{R}_{\geq 0}$, we have

$$||\mathbf{e}_{n+1}||_2 \leq ||\mathbf{e}_n||_2 + \mathcal{O}(L(\sigma(n+1) - \sigma(n))^{p+1})$$

For the base case of $\mathbf{e}_\epsilon$, $f_\theta(\mathbf{x}_\epsilon) = \mathbf{x}_\epsilon$ as we assume $\mathcal{L}_{\text{Recon}}(\theta) = 0$ and by definition, $f(\mathbf{x}_\epsilon) = \mathbf{x}_\epsilon$, and thus we have $\mathbf{e}_\epsilon = 0$.

We can now bound the error $||\mathbf{e}_n||_2$, by induction on the error of previous noise levels,

$$\begin{aligned}
||\mathbf{e}_n||_2 &\leq L||\hat{\mathbf{x}}_{t_{n-1}} - \mathbf{x}_{t_{n-1}}||_2 + ||\mathbf{e}_{n-1}||_2 && (16)\\
&= \sum_{i=\epsilon}^{n-1} L\mathcal{O}((\sigma(t_{i+1}) - \sigma(t_i)^{p+1}) && (17)\\
&\leq \sum_{i=\epsilon}^{n-1} (\sigma(t_{i+1}) - \sigma(t_i))(L\mathcal{O}(\Delta)^p). && (18)\\
&= (L\mathcal{O}(\Delta)^p)(t_n - \epsilon) && (19)
\end{aligned}$$

As $t_n - \epsilon \leq t_N - \epsilon \leq C$, where C is some constant. We therefore have,

$$(L\mathcal{O}(\Delta)^p)(t_n - \epsilon) \leq C(\mathcal{O}(\Delta)^p). \quad (20)$$

As C and L are constants and can be neglected compared to the exponential term, we have $C(L\mathcal{O}(\Delta)^p) = (\mathcal{O}(\Delta)^p)$, which completes the proof. □

C IGN training

C.1 IGN training pseudocode

To be self-consistent, we reproduce the training procedure for a standard IGN.

D Experiment details

Model Architecture Intuitively, we attempt to parameterize the model based on existing work on the consistency and diffusion models. We based our model on the same architecture as the pre-trained model, with our custom loss functions employed.

Algorithm 4 Training an idempotent generative network

Input: Data set $\mathcal{D}$, models ϕ_θ, $\phi_{\theta^`}$, drift measure $\delta(\cdot, \cdot)$
while not converged **do**
  Sample $x \sim \mathcal{D}$
  Sample $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Copy $\phi_{\theta^`} \leftarrow \phi_\theta$
  $x_{\text{recon}}, x_{\text{sample}} \leftarrow \phi_\theta(x), \phi_\theta(z)$
  $x_{\text{idem}} \leftarrow \phi_{\theta^`}(z)$
  $x_{\text{tight}} \leftarrow \phi_{\theta^`}(z)$
  Copy $x_{\text{clone}} \leftarrow x_{\text{sample}}$
  $x_{\text{proj}} \leftarrow \phi_\theta(x_{\text{clone}})$
  $\mathcal{L}_{\text{recon}} \leftarrow \delta(x, x_{\text{recon}})$
  $\mathcal{L}_{\text{idem}} \leftarrow \delta(x_{\text{sample}}, x_{\text{idem}})$
  $\mathcal{L}_{\text{tight}} \leftarrow -\delta(x_{\text{proj}}, x_{\text{clone}})$
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{recon}} + \lambda_i \mathcal{L}_{\text{idem}} + \lambda_t \mathcal{L}_{\text{tight}}$
  $\phi_\theta \leftarrow \phi_\theta - \eta \nabla_\theta(\phi_\theta)$
end while

Training details We trained them on a system with 4 Nvidia H100 GPUs, using PyTorch as the framework. Since the SIGN contains a subset of the parameters of the diffusion model, we initialize the SIGN using the parameters of a trained diffusion model. Unless otherwise specified, all hyperparameters were identical to those of the respective base models. We use the original noise schedule from EDM and DDIM respectively to train our SIGN models. We also employed techniques from Distribution Matching DistillationYin et al. [2024a] of using a pre-generated image to help our model get to the target manifold faster. For CIFAR-10, we started from Karras et al. [2022], with 200K pre-generated samples. For CelebA, we based our work on Song et al. [2020b], with 500K pre-generated samples and trained for 350 epochs. To further ensure training stability, we initialized our models with pre-trained weights.

Evaluation Setup For each dataset, we generated 50K unconditioned single-step samples from our trained model and used FID to evaluate their overall likeliness to the target dataset.

Source Code The source code and instructions to run the experiment can be acquired through this link: https://anonymous.4open.science/r/SIGN-88/

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main claim made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In section 7 we acknowledges the limitations of this work as a proof of concept, and its lower performance compared to existing generative models.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Our theory proofs are detailed in Appendix B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experiment settings are detailed in both Section 6 and Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The source code as well as instruction on how to run are provided.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experiment settings are detailed in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the Fréchet Inception Distance (FID) between 50,000 generated samples for each target dataset. This sample size is used to ensure a stable and low-bias estimation of the FID score.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The experiment settings are detailed in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed. Our work here is merely a proof of concept for a foundational algorithm, and not tied to any specific applications.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The datasets and models used in this paper are properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLM usages is not involved in this research.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.