Fast Sampling of Diffusion Models via Operator Learning

Hongkai Zheng - Weilie Nie - Arash Vahdat - Kamyar Azizzadenesheli - Anima Anandkumar
Caltech - NVIDIA

Proceedings of the 40th International Conference on Machine Learning - July 2023

Presented by: Amirhossein Alimohammadi

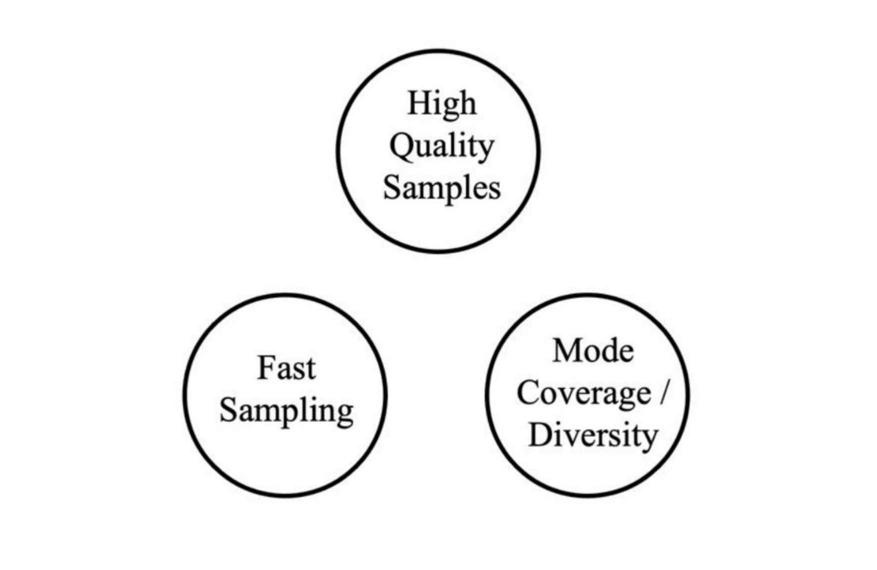
Main Objective

Diffusion models are utilized in various areas.

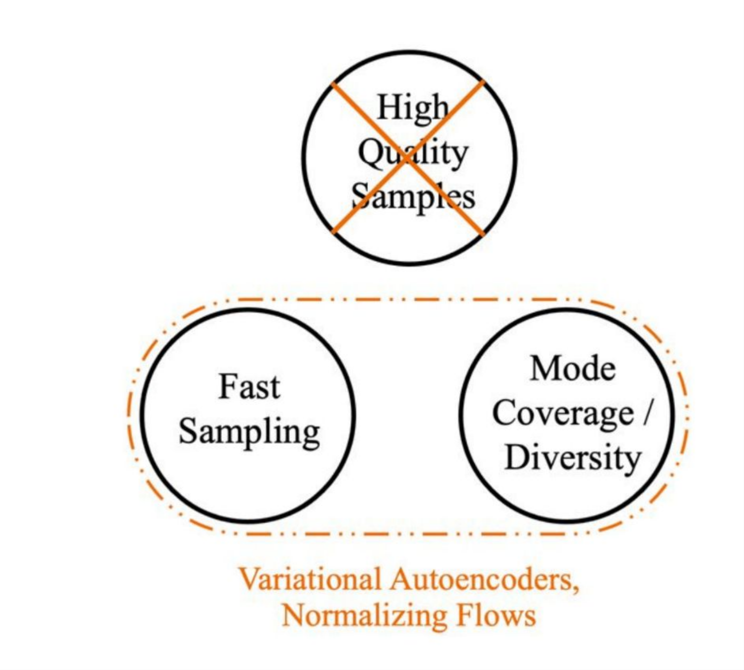
However, they carry a drawback!!!!

Their sampling process is slow because it requires hundreds to thousands of network evaluations to emulate a continuous process.

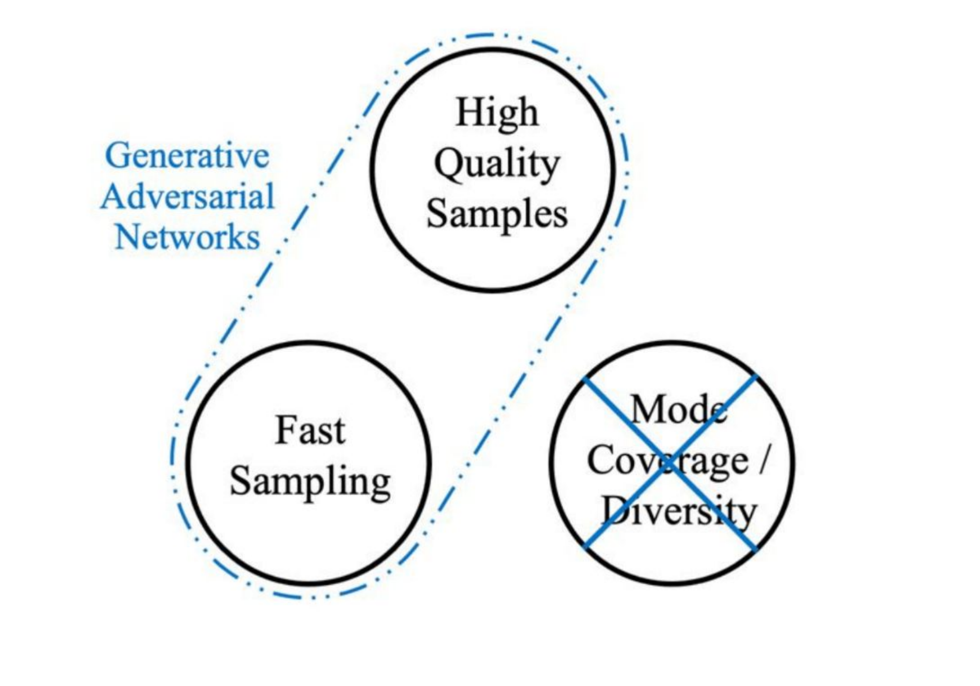
The Generative Learning Trilemma



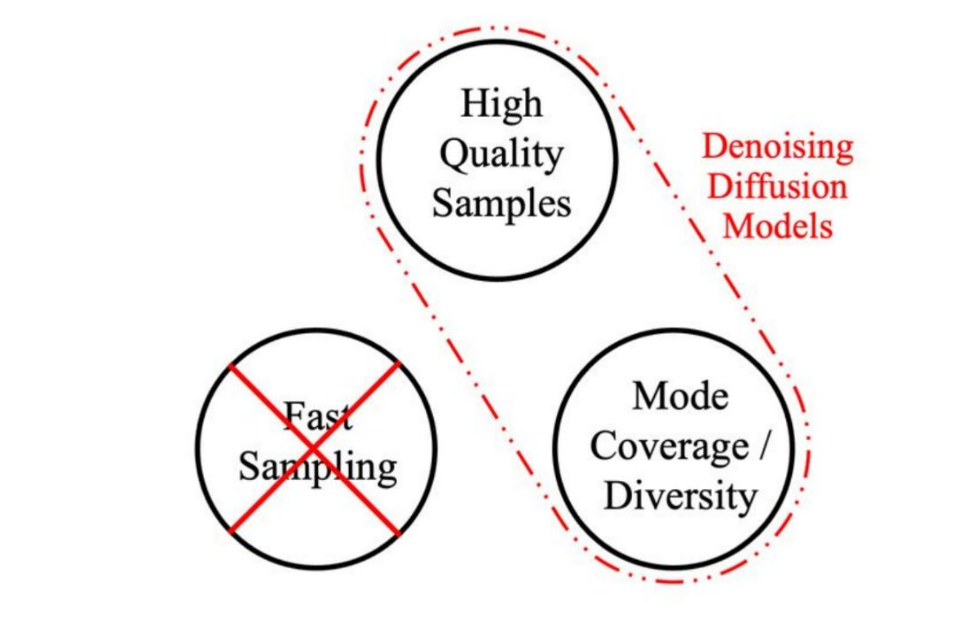
The Generative Learning Trilemma


~~High Quality Samples~~
Fast Sampling
Mode Coverage / Diversity
Variational Autoencoders, Normalizing Flows

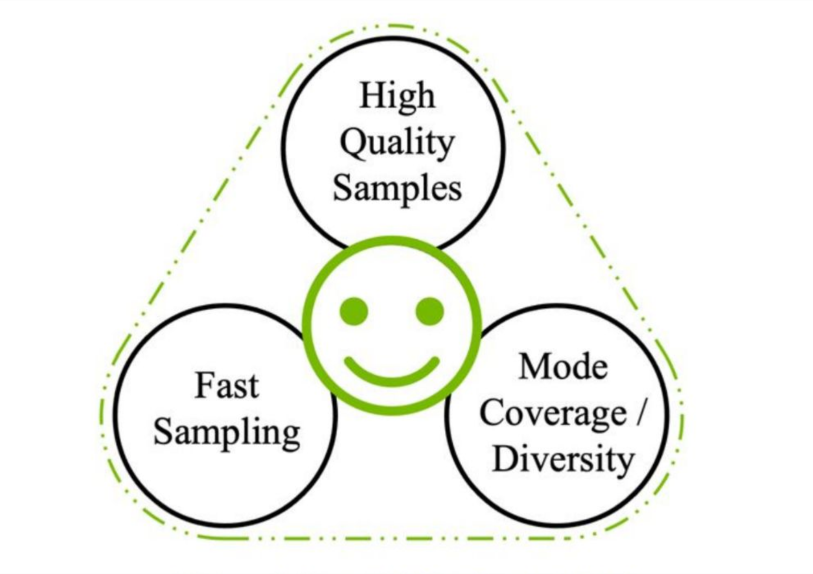
The Generative Learning Trilemma



The Generative Learning Trilemma



Diffusion Model Sampling with Neural Operator (DSNO)



Previous Work

Method	Number of Function Evaluations (NFE)
Knowledge distillation (Luhman & Luhman, 2021)	1
LSGM (Vahdat et al., 2021)	147
DDIM (Song et al., 2020a)	10 - 20 - 50
FastDPM (Kong & Ping, 2021)	10
DDGAN (Xiao et al., 2021)	4

Why DSNO outperforms previous methods?

1. It uses only 1 NFE.

2. It has a significant improvement compare to the models working with at most 5 NFE.

3. It can be built upon any existing architecture of diffusion models.

Background - Neural Operator

What is a Neural Operator?

- A neural operator is a type of neural network architecture designed to learn and approximate mathematical operators.
- It aims to model the mapping between input and output spaces, capturing complex relationships and patterns.

Neural Network vs. Neural Operator

The classical development of neural networks:

- Mappings between a finite-dimensional Euclidean space and a set of classes, or between two finite-dimensional Euclidean spaces.

Neural Operator:

- Mappings between infinite-dimensional spaces (operators).

Fourier Neural Operator

They formulate a new neural operator by parameterizing the integral kernel directly in Fourier space. (Yang et al., 2021; Wen et al., 2022)

Benefits of this method:

1. They provide a resolution-invariant solution. So they can be used to do zero-shot super-resolution: trained on a lower resolution directly and evaluated on a higher resolution.

2. One of the most efficient machine learning methods for scientific computing problems involving PDE. (On a 256 * 256 grid, the inference time of FNO is 0.005s compared to the 2.2s of the pseudo-spectral method)

Learning the trajectory with neural operator

Our objective:

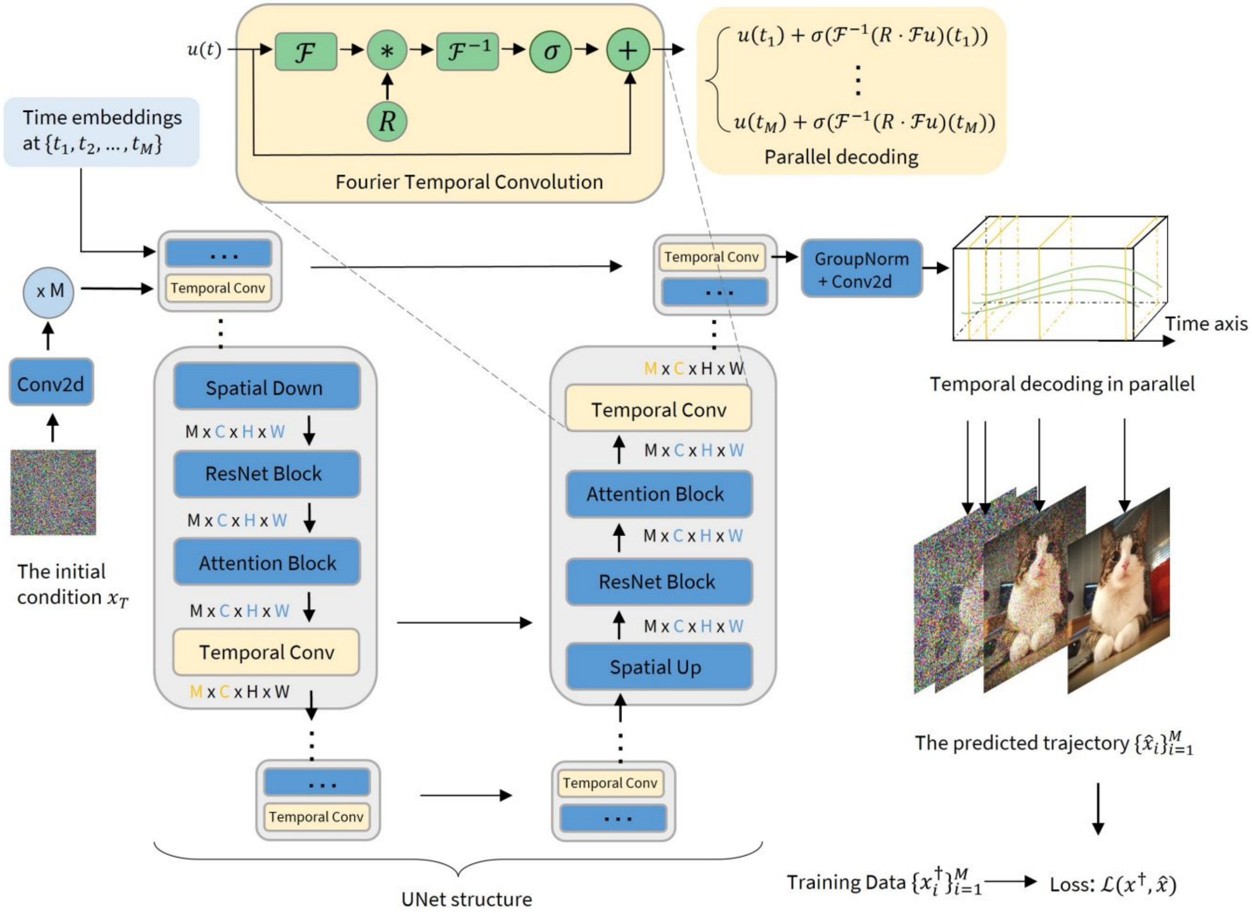
Learn the solution operator for a diffusion process. (mapping an initial condition to the probability flow trajectory over time)

Architecture

1. DSNO is built on an existing diffusion model (backbone) and includes temporal convolution layers added to the U-Net structure.

2. The temporal convolutions operate on the temporal and channel dimensions, making the architecture highly parallelizable with minimal computational complexity.

3. The addition of temporal convolutions enhances the model's ability to capture temporal dependencies in diffusion processes.


$u(t)$
$\mathcal{F}$
$*$
$\mathcal{F}^{-1}$
σ
$+$
R
Fourier Temporal Convolution
$u(t_1) + \sigma(\mathcal{F}^{-1}(R \cdot \mathcal{F}u)(t_1))$
$\vdots$
$u(t_M) + \sigma(\mathcal{F}^{-1}(R \cdot \mathcal{F}u)(t_M))$
Parallel decoding
Time embeddings at $\{t_1, t_2, \ldots, t_M\}$
x M
Conv2d
The initial condition x_T
. . .
Temporal Conv
Spatial Down
M x C x H x W
ResNet Block
M x C x H x W
Attention Block
M x C x H x W
Temporal Conv
M x C x H x W
. . .
Temporal Conv
Temporal Conv
. . .
M x C x H x W
Temporal Conv
M x C x H x W
Attention Block
M x C x H x W
ResNet Block
M x C x H x W
Spatial Up
Temporal Conv
. . .
UNet structure
GroupNorm + Conv2d
Time axis
Temporal decoding in parallel
The predicted trajectory $\{\hat{x}_i\}_{i=1}^{M}$
Training Data $\{x_i^\dagger\}_{i=1}^{M}$
Loss: $\mathcal{L}(x^\dagger, \hat{x})$

Loss Function

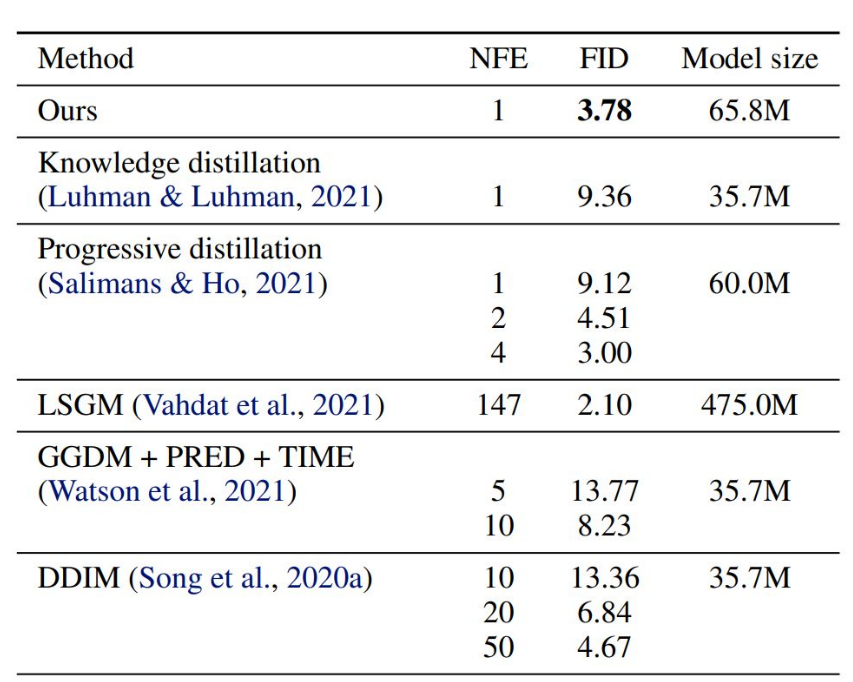
Objective: Minimize empirical risk

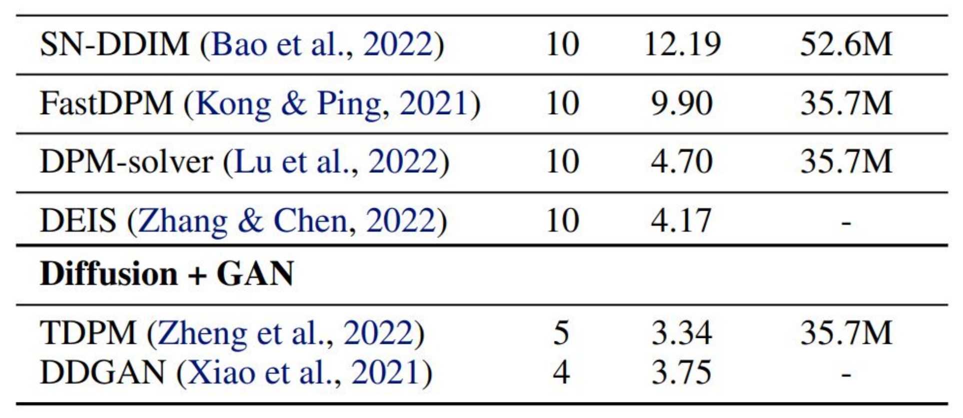
Empirical risk is measured by comparing DSNO predictions with the true solution operator.

A weighted integral of error with a weighting function that accounts for the importance of different time points.

$$\min_{\theta} \frac{1}{N} \sum_{j=1}^{N} \frac{1}{M} \sum_{i=1}^{M} \lambda\left(t_i\right) \left\| \mathcal{G}_\theta\left(\mathbf{x}_T^{(j)}\right)\left(t_i\right) - \mathcal{G}^\dagger\left(\mathbf{x}_T^{(j)}\right)\left(t_i\right) \right\|$$

Experiments

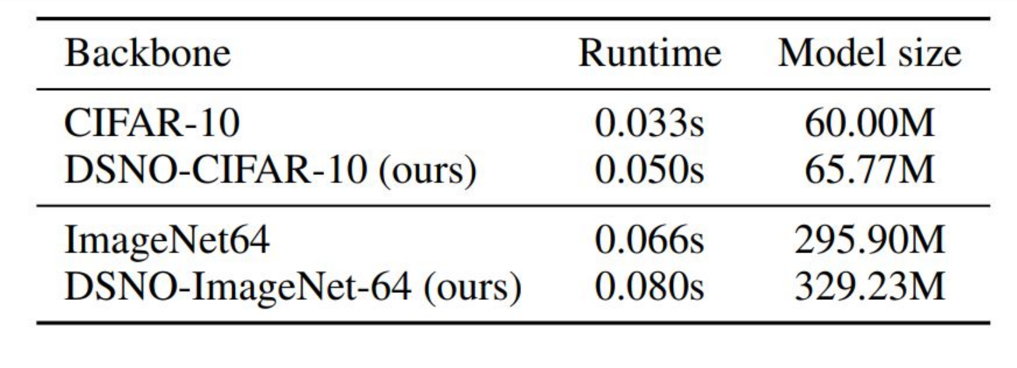




Method	NFE	FID	Model size
Ours	1	**3.78**	65.8M
Knowledge distillation (Luhman & Luhman, 2021)	1	9.36	35.7M
Progressive distillation (Salimans & Ho, 2021)	1	9.12	60.0M
	2	4.51	
	4	3.00	
LSGM (Vahdat et al., 2021)	147	2.10	475.0M
GGDM + PRED + TIME (Watson et al., 2021)	5	13.77	35.7M
	10	8.23	
DDIM (Song et al., 2020a)	10	13.36	35.7M
	20	6.84	
	50	4.67	
SN-DDIM (Bao et al., 2022)	10	12.19	52.6M
FastDPM (Kong & Ping, 2021)	10	9.90	35.7M
DPM-solver (Lu et al., 2022)	10	4.70	35.7M
DEIS (Zhang & Chen, 2022)	10	4.17	-
Diffusion + GAN			
TDPM (Zheng et al., 2022)	5	3.34	35.7M
DDGAN (Xiao et al., 2021)	4	3.75	-

Comparison of fast sampling methods on CIFAR-10 for diffusion models in the literature. T NFE: number of function evaluations.

Experiments

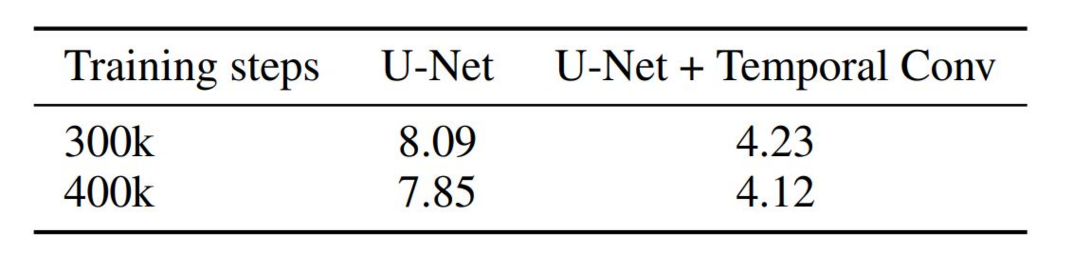


Backbone	Runtime	Model size
CIFAR-10	0.033s	60.00M
DSNO-CIFAR-10 (ours)	0.050s	65.77M
ImageNet64	0.066s	295.90M
DSNO-ImageNet-64 (ours)	0.080s	329.23M

One model evaluation cost tested on V100. The time cost of a single forward pass of DSNO and the corresponding original backbone are being compared. The reported results are averaged over 20 runs. The baseline models are from Salimans & Ho (2021).

Experiments

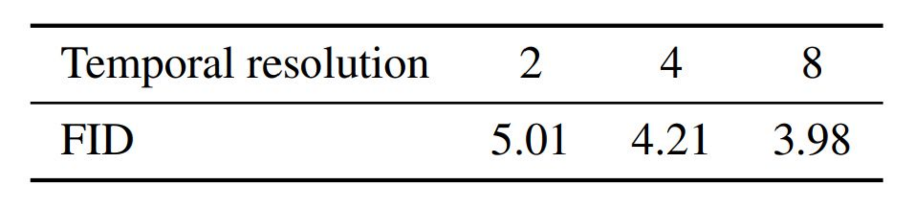
Ablation study on the impact of using the temporal convolution.



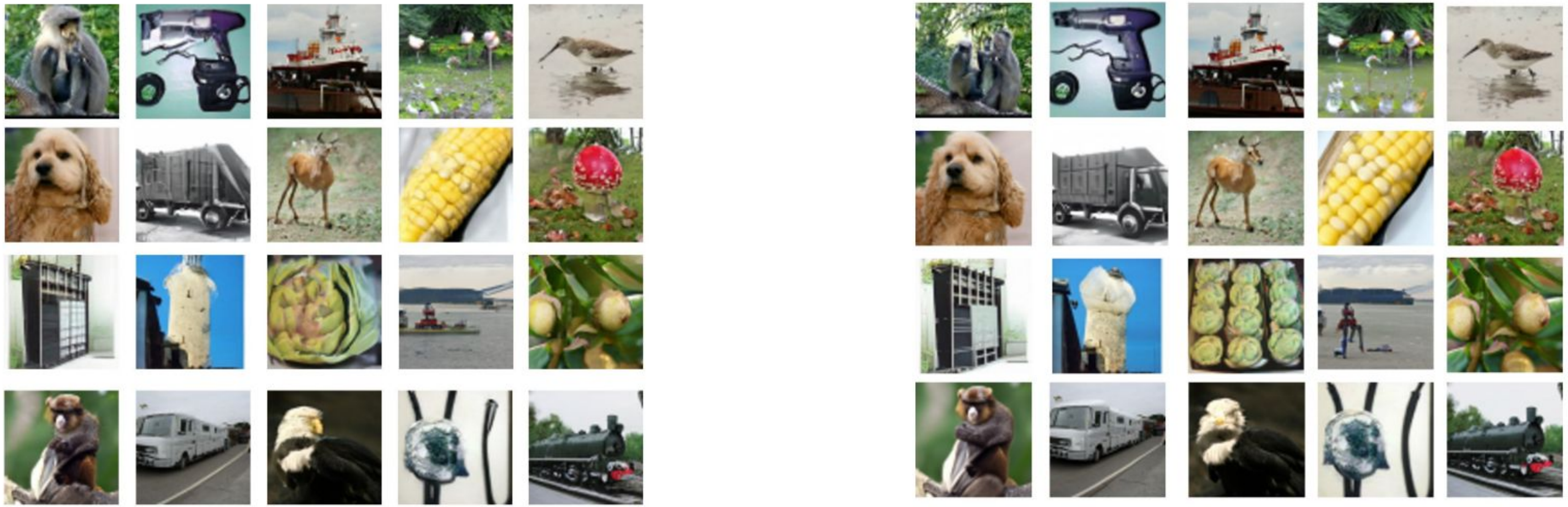
Training steps	U-Net	U-Net + Temporal Conv
300k	8.09	4.23
400k	7.85	4.12

Experiments

Ablation study on the choice of using the temporal resolution.



Temporal resolution	2	4	8
FID	5.01	4.21	3.98



The figure shows the random samples from DSNO and the original pre-trained diffusion model with the same random seed. (Left panel: random samples generated by DSNO. Right panel: generated by the solver.)

Thank you!

- Thank you for your attention!
- I appreciate your time and interest.
- If you have any questions, please feel free to ask.
- Contact information: alimohammadiamirhossein@gmail.com

Mathematical Slides

How to map the operators?

Let $D \subset \mathbb{R}^d$be a bounded, open set and $\mathcal{A} = \mathcal{A}\left(D; \mathbb{R}^{d_a}\right)$and $\mathcal{U} = \mathcal{U}\left(D; \mathbb{R}^{d_u}\right)$ be Banach spaces of function taking values in $\mathbb{R}^{d_a}$ and $\mathbb{R}^{d_u}$respectively.

Furthermore, $G^\dagger : \mathcal{A} \to \mathcal{U}$ let be a (typically) non-linear map. Suppose we have observations $\{a_j, u_j\}_{j=1}^N$ where $a_j \sim \mu$is an i.i.d. sequence from the probability measure μ supported on $\mathcal{A}$ and $u_j = G^\dagger(a_j)$is possibly corrupted with noise. We aim to build an approximation of $G^\dagger$ by constructing a parametric map

$$G : \mathcal{A} \times \Theta \to \mathcal{U} \quad \text{or equivalently,} \quad G_\theta : \mathcal{A} \to \mathcal{U}, \quad \theta \in \Theta$$

How to map the operators?

Iterative Updates: $v_{t+1}(x) := \sigma\left(W v_t(x) + \left(\mathcal{K}(a;\phi)v_t\right)(x)\right), \quad \forall x \in D$

where $\mathcal{K} : \mathcal{A} \times \Theta_{\mathcal{K}} \to \mathcal{L}\left(\mathcal{U}\left(D;\mathbb{R}^{d_v}\right), \mathcal{U}\left(D;\mathbb{R}^{d_v}\right)\right)$ maps to bounded linear operators on $\mathcal{U}\left(D;\mathbb{R}^{d_v}\right)$

Kernel Integral Operator $\left(\mathcal{K}(a;\phi)v_t\right)(x) := \int_D \kappa(x, y, a(x), a(y); \phi) v_t(y) \mathrm{d}y, \quad \forall x \in D$

The output is: $u(x) = Q\left(v_T(x)\right)$

Fourier Transform

Let $\mathcal{F}$ denote the Fourier transform of a function $f : D \to \mathbb{R}^{d_v}$ and $\mathcal{F}^{-1}$ its inverse then $(\mathcal{F}f)_j(k) = \int_D f_j(x) e^{-2i\pi\langle x,k \rangle} \mathrm{d}x, \quad \left(\mathcal{F}^{-1}f\right)_j(x) = \int_D f_j(k) e^{2i\pi\langle x,k \rangle} \mathrm{d}k$

For $j = 1, \ldots, d_v$ where $i = \sqrt{-1}$ is the imaginary unit.
By letting $\kappa_\phi(x, y, a(x), a(y)) = \kappa_\phi(x - y)$ in last slide and applying the convolution theorem, we find that $\left(\mathcal{K}(a;\phi)v_t\right)(x) = \mathcal{F}^{-1}\left(\mathcal{F}\left(\kappa_\phi\right) \cdot \mathcal{F}\left(v_t\right)\right)(x), \quad \forall x \in D$

We, therefore, propose to directly parameterize κ_ϕ in Fourier space.

Fourier Transform

Define the Fourier integral operator $\left(\mathcal{K}(\phi)v_t\right)(x) = \mathcal{F}^{-1}\left(R_\phi \cdot (\mathcal{F}v_t)\right)(x) \quad \forall x \in D$.

where R_ϕ is the Fourier transform of a periodic function $\kappa : \bar{D} \to \mathbb{R}^{d_v \times d_v}$ parameterized by $\phi \in \Theta_{\mathcal{K}}$.

We have $(\mathcal{F}v_t)(k) \in \mathbb{C}^{d_v}$ and $R_\phi(k) \in \mathbb{C}^{d_v \times d_v}$. Notice that since we assume κ is periodic, it admits a Fourier series expansion, so we may work with the discrete modes $k \in \mathbb{Z}^d$.

We pick a finite-dimensional parameterization by truncating the Fourier series at a maximal number of modes $k_{\max} = |Z_{k_{\max}}| =| \left\{k \in \mathbb{Z}^d : |k_j| \leq k_{\max,j}\right.$, for $\left. j = 1, \ldots, d\right\} |$. We thus parameterize R_ϕ directly as complex-valued tensor $(k_{\max} \times d_v \times d_v)$ comprising a collection of truncated Fourier modes.

Fourier Transform

A Fourier neural operator $\mathcal{G}_\theta$, parameterized with learnable parameters θ ,

is an $\mathcal{L}$ layered neural operator of the following form,

$$\mathcal{G}_\theta := \mathcal{Q} \circ \sigma\left(\mathcal{W}_L + \mathcal{K}_L\right) \circ \cdots \circ \sigma\left(\mathcal{W}_1 + \mathcal{K}_1\right) \circ \mathcal{P}$$

where the lifting operator $\mathcal{P}$, projection operator $\mathcal{Q}$, and residual connections

$\mathcal{W}_i, i \in \{1, \ldots, L\}$ are pointwise operators parameterized with neural networks, and

σ is a fixed nonlinear activation function. $\mathcal{K}_i$ is an integral kernel operator.

DSNO

The aim is approximating a solution operator by minimizing the following error

$$\min_{\theta} \mathbb{E}_{\mathbf{x}_T \sim \mathcal{N}(0,\mathbf{I})} \mathcal{L}\left(\mathcal{G}_\theta\left(\mathbf{x}_T\right) - \mathcal{G}^\dagger\left(\mathbf{x}_T\right)\right)$$

Where $\mathcal{L} : \mathcal{U} \to \mathbb{R}_+$ is some loss functional such as L^p norm for some $p \geq 1$.

Temporal convolution block in Fourier space

There is an equation that resembles the one in Fourier Transform, but it includes specific adjustments designed to enhance its suitability for the particular problem.

$$(\mathcal{T}u)(t) = u(t) + \sigma((\mathcal{K}u)(t))$$

Where $(\mathcal{K}u)(t) = \int_D \left(\mathcal{F}^{-1}R\right)(\tau)u(t-\tau)\mathrm{d}\tau, \forall t \in D.$

The pointwise product of Fourier transforms

$$R \cdot (\mathcal{F}u)_{j,k} = \sum_{l=1}^{d} R_{j,k,l}(\mathcal{F}u)_{j,l}$$

Where $\mathcal{J}$ is the number of modes and d is the number of hidden dimensions.

What is mode?

The discrete-time Fourier transform of the signal $x(t)$ with period T is given by

$$X_j = \sum_{i=1}^{N} x\left(t_i\right) \exp\left(-\frac{2\pi}{T} j i t_i\right),$$

where $t_i = \frac{iT}{N} \cdot \frac{j}{T}$ is the frequency. j is called the frequency mode. Let $\Delta = \frac{1}{N}$ be the time step.