Poisson Flow Generative Models

Yilun Xu*, Ziming Liu*, Max Tegmark, Tommi Jaakkola

NeurIPS 2022

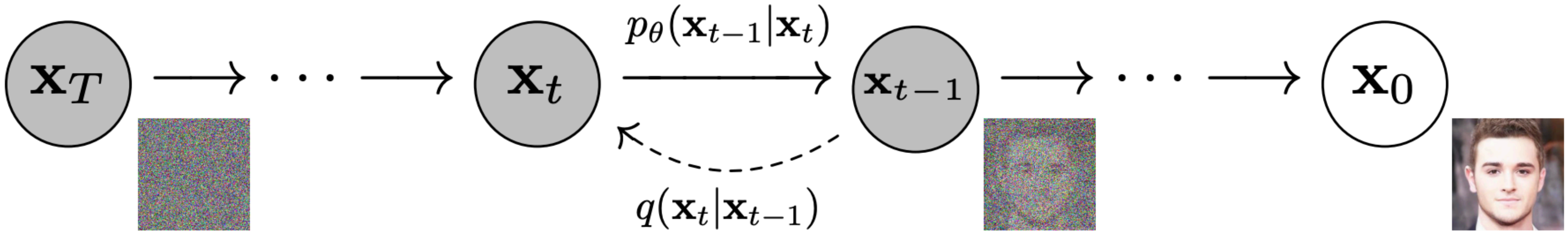
STRUCT Group Seminar
Presenter: Wenjing Wang
2022.10.30

OUTLINE

- Authorship
- Background
- Method
- Experiments
- Conclusion

BACKGROUND

- Denoising diffusion probabilistic model



BACKGROUND

➤ Denoising diffusion probabilistic model

- Noising process:

$$x_i = \sqrt{1-\beta_i}\,x_{i-1} + \sqrt{\beta_i}\,z_{i-1}, \qquad z_{i-1} \sim^{iid} \mathcal{N}(0,\mathrm{I})$$

- T-time conditional transition:

$$q(x_t \mid x_0) = \mathcal{N}(x_t;\ \sqrt{\bar{\alpha}_t}x_0,\ (1-\bar{\alpha}_t)\mathbf{I})$$

$$\text{where }\ \alpha_t = 1-\beta_t\ \text{ and }\ \bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$$

BACKGROUND

➤ Score-based generative model

- DDPM's noising process:

$$x_i = \sqrt{1-\beta_i}x_{i-1} + \sqrt{\beta_i}z_{i-1}, \qquad z_{i-1} \sim^{iid} \mathcal{N}(0,\mathrm{I})$$

- Markov Chian turns to stochastic process if time continues:

$$dX_t = -\frac{1}{2}\beta(t)X_t dt + \sqrt{\beta(t)}dW_t$$

where W is an independent Weiner process

BACKGROUND

➤ Score-based generative model

- DDPM's t-time conditional transition:

$$q(x_t | x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t)\mathbf{I})$$

- Markov Chian turns to stochastic process if time continues:

$$q_{0_t}(x(t) | x(0)) = \mathcal{N}(x(t); \ x(0)e^{-\frac{1}{2}\int_0^t \beta(s)ds}, \mathbf{I} - \mathbf{I}e^{-\int_0^t \beta(s)ds})$$

If $\int_0^t \beta(s)ds \to +\infty, t \to T$ then $q_{0_t}(x(t) | x(0)) \to \mathcal{N}(x(t); 0, \ \mathbf{I})$

BACKGROUND

- Score-based generative model
  - How to reverse above stochastic process?
  - Consider a general forward diffusion process,

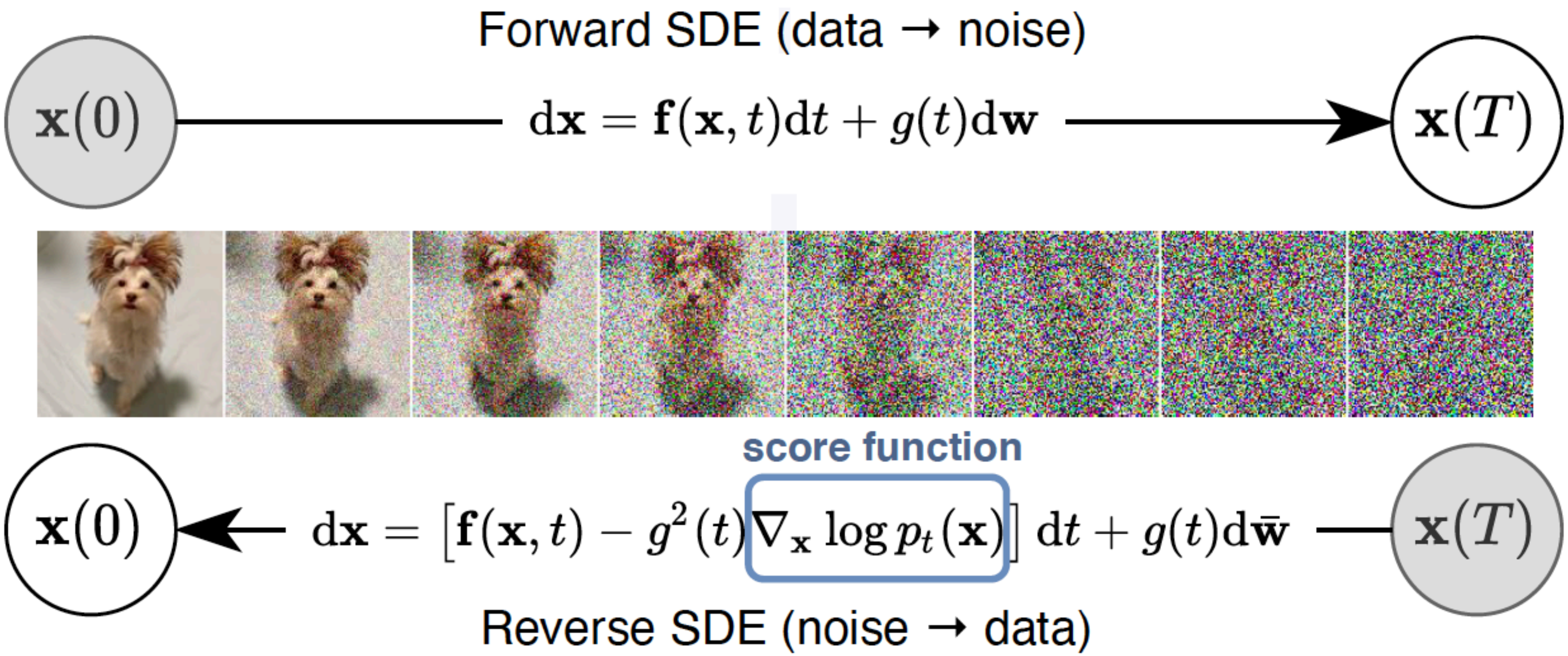
$$dX_t = f(X_t, t)dt + g(t)dW_t$$

  - There exits a reverse stochastic process which share the same marginal distribution as the forward one

$$dX_t = [f(X_t, t) - g(t)^2 \nabla_{x_t} \log p_t(X_t)]dt + g(t)d\bar{W}_t$$

BACKGROUND

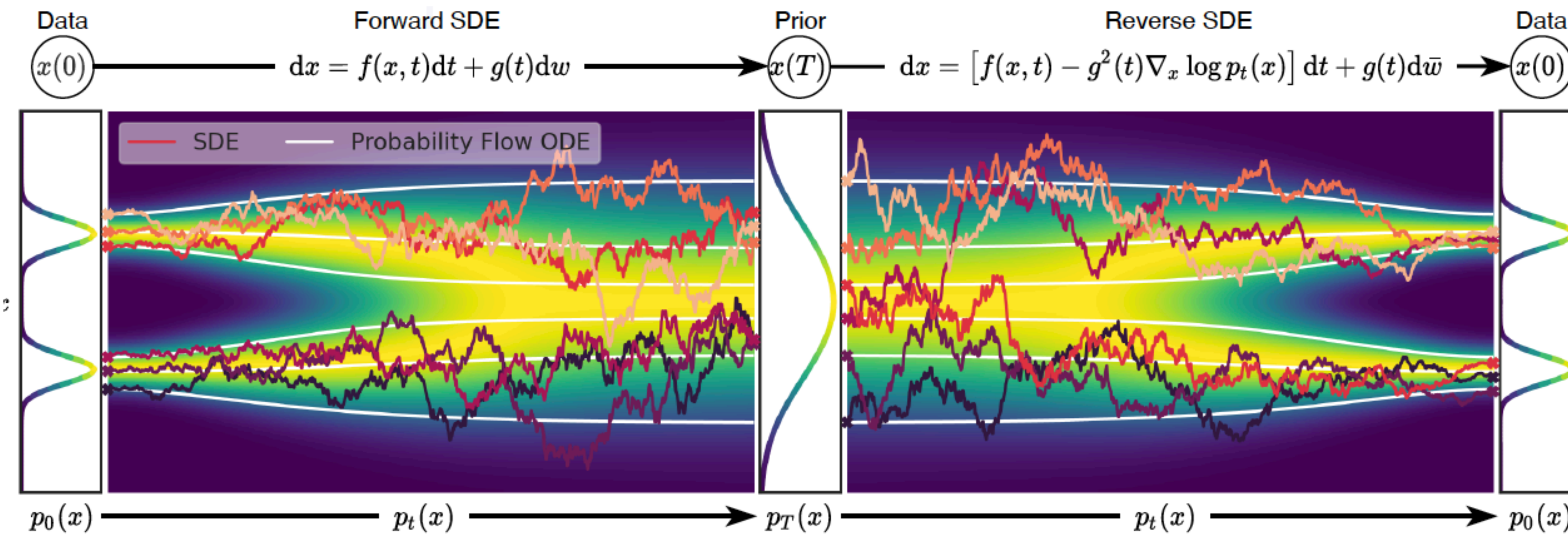
- Score-based generative model



Score-Based Generative Modeling Through Stochastic Differential Equations (ICLR-21)

BACKGROUND

- Score-based generative model



Score-Based Generative Modeling Through Stochastic Differential Equations (ICLR-21)

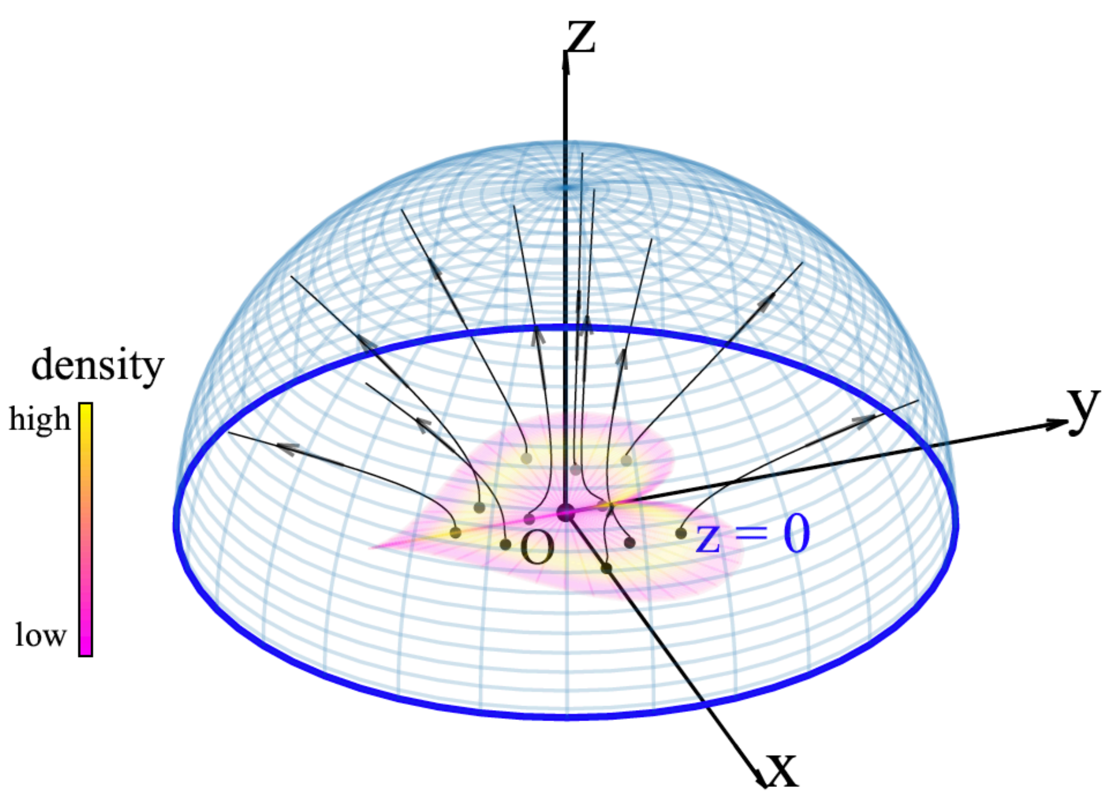
OUTLINE

- Authorship
- Background
- Method
- Experiments
- Conclusion

METHOD

➤ "Poisson flow" generative model (PFGM)

- Motion in a viscous fluid transforms any planar charge distribution into a uniform angular distribution.

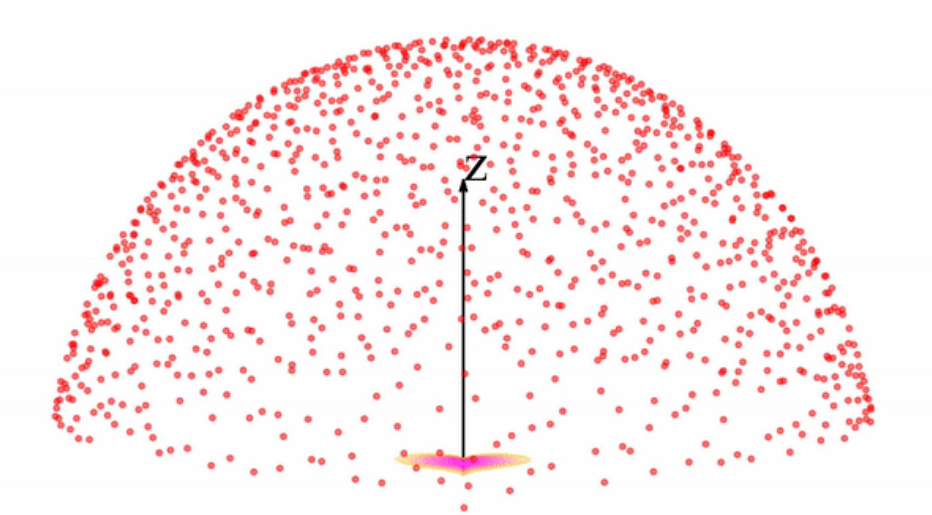


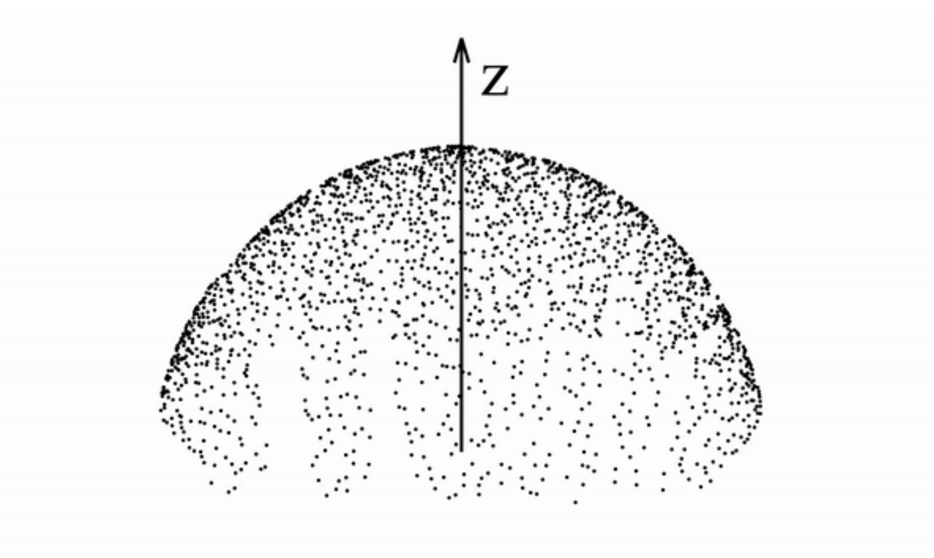
A positive charge with $z > 0$, move in the direction of their repulsive force, eventually crossing an imaginary hemisphere of radius r.

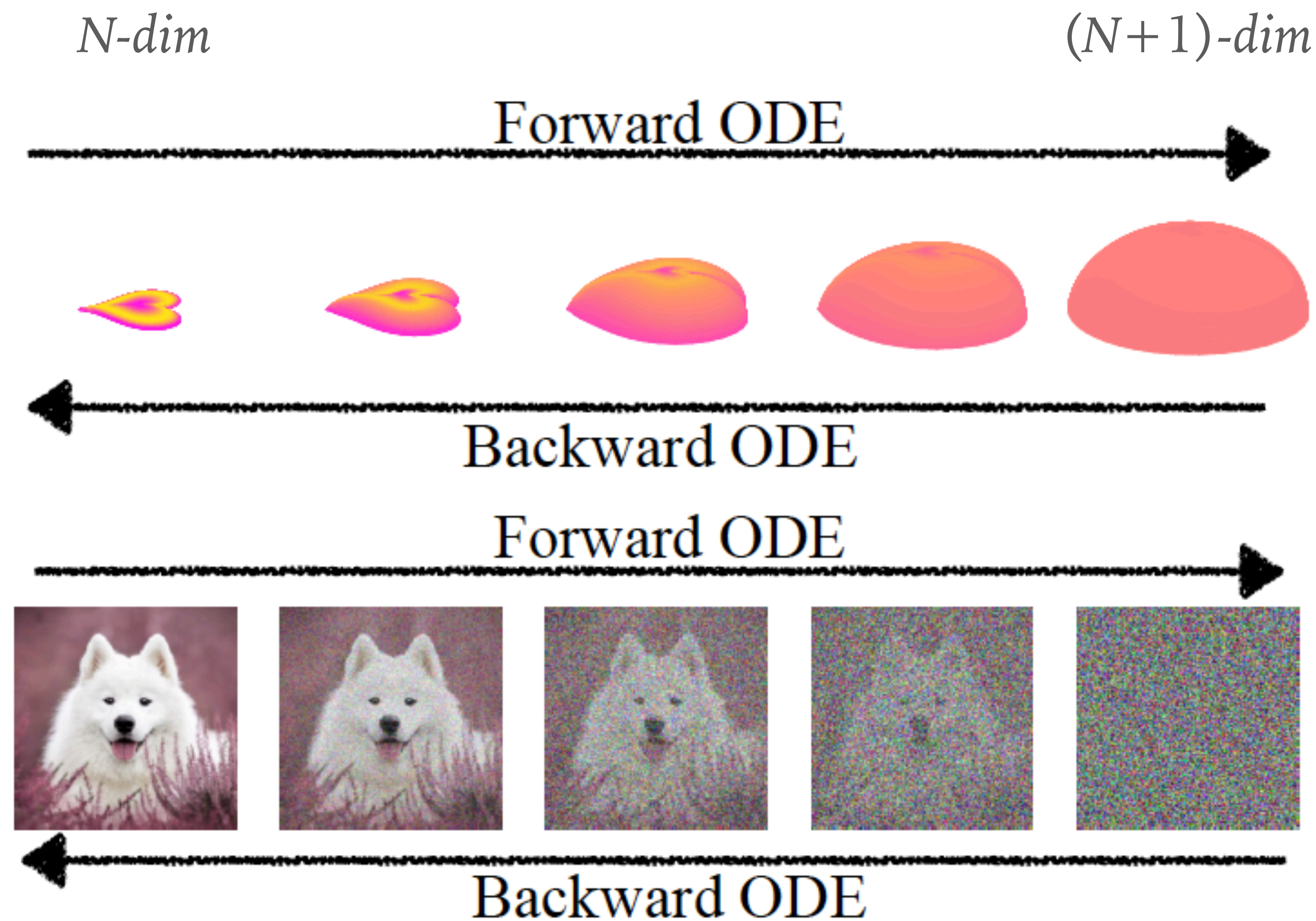
If the the original charge distribution is let loose just above $z = 0$, it will cause a uniform distribution for $r \to \infty$.

METHOD

- “Poisson flow” generative model (PFGM)







METHOD - BACKGROUND

- Poisson equation

$$\Delta\varphi = f$$

where Δ is the Laplace operator

f and φ are real or complex-valued functions on a manifold

- Usually, f is given and φ is sought

- Euclidean space: $\nabla^2\varphi = f$

METHOD - BACKGROUND

➤ Poisson equation

Poisson equation Let $\mathbf{x} \in \mathbb{R}^N$ and $\rho(\mathbf{x}) : \mathbb{R}^N \to \mathbb{R}$ be a *source* function. We assume that the source function has a compact support, $\rho \in \mathcal{C}^0$ and $N \geq 3$. The Poisson equation is

$$\nabla^2 \varphi(\mathbf{x}) = -\rho(\mathbf{x}), \tag{1}$$

where $\varphi(\mathbf{x}) : \mathbb{R}^N \to \mathbb{R}$ is called the *potential function*, and $\nabla^2 \equiv \sum_{i=1}^N \frac{\partial^2}{\partial x_i^2}$ is the Laplacian operator. It is usually helpful to define the gradient field $\mathbf{E}(\mathbf{x}) = -\nabla \varphi(\mathbf{x})$ and rewrite the Poisson equation as $\nabla \cdot \mathbf{E} = \rho$, known in physics as Gauss's law [11].

METHOD - BACKGROUND

➤ Poisson equation

$$\nabla^2 \varphi(\mathbf{x}) = -\rho(\mathbf{x}), \tag{1}$$

- With zero boundary condition at infinity, Eq. (1) admits a unique simple integral solution

$$\varphi(\mathbf{x}) = \int G(\mathbf{x}, \mathbf{y})\rho(\mathbf{y})d\mathbf{y}, \quad G(\mathbf{x}, \mathbf{y}) = \frac{1}{(N-2)S_{N-1}(1)} \frac{1}{||\mathbf{x}-\mathbf{y}||^{N-2}}, \tag{2}$$

where $S_{N-1}(1)$ is a geometric constant representing the surface area of the unit $(N-1)$-sphere [3], and $G(\mathbf{x}, \mathbf{y})$ is the extension of Green's function in N-dimensional space (details in Appendix A.3). The negative gradient field of $\varphi(\mathbf{x})$, referred as *Poisson field* of the source ρ, is

$$\mathbf{E}(\mathbf{x}) = -\nabla\varphi(\mathbf{x}) = -\int \nabla_{\mathbf{x}} G(\mathbf{x}, \mathbf{y})\rho(\mathbf{y})d\mathbf{y}, \quad \nabla_{\mathbf{x}} G(\mathbf{x}, \mathbf{y}) = -\frac{1}{S_{N-1}(1)} \frac{\mathbf{x}-\mathbf{y}}{||\mathbf{x}-\mathbf{y}||^{N}}. \tag{3}$$

METHOD - BACKGROUND

$$\mathbf{E}(\mathbf{x}) = -\nabla\varphi(\mathbf{x}) = -\int \nabla_{\mathbf{x}} G(\mathbf{x},\mathbf{y})\rho(\mathbf{y})d\mathbf{y},$$

$$\nabla_{\mathbf{x}} G(\mathbf{x},\mathbf{y}) = -\frac{1}{S_{N-1}(1)}\frac{\mathbf{x}-\mathbf{y}}{||\mathbf{x}-\mathbf{y}||^N}.$$

- $-\mathbf{E}(\mathrm{x})$ points towards sources

It is straightforward to check that when $\rho(\mathbf{x}) \to \delta(\mathbf{x}-\mathbf{y})$, we get $\varphi(\mathbf{x}) \to G(\mathbf{x},\mathbf{y})$ and $\mathbf{E}(\mathbf{x}) \to -\nabla_{\mathbf{x}} G(\mathbf{x},\mathbf{y})$. This implies that $G(\mathbf{x},\mathbf{y})$ and $-\nabla_{\mathbf{x}} G(\mathbf{x},\mathbf{y})$ can be interpreted as the potential function and the gradient field generated by a unit point source, *e.g.*, a point charge, located at $\mathbf{y}$. When $\rho(\mathbf{x})$ takes general forms but has bounded support, simple asymptotics exist for $||\mathbf{x}|| \gg ||\mathbf{y}||$. To the lowest order, $\mathbf{E}(\mathbf{x}) = \nabla_{\mathbf{x}} G(\mathbf{x},\mathbf{y})|_{\mathbf{y}=\mathbf{0}} \sim \mathbf{x}/||\mathbf{x}||^N$ behaves as if it were generated by a unit point source at $\mathbf{y} = 0$. In physics, the power law decay is considered to be long-range (compared to exponential decay) [11].

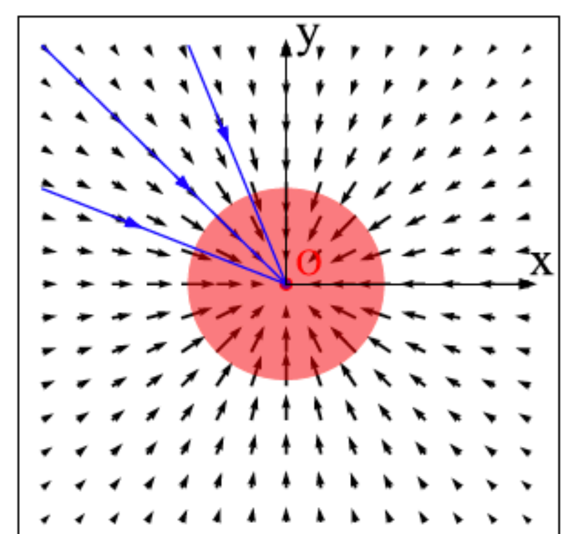
METHOD - HOW TO DRAW SAMPLES

- Since $-E(x)$ points towards sources, the backward ODE $\frac{\mathrm{d}x}{\mathrm{d}t} = -E(x)$ will take samples close to the sources.

METHOD - HOW TO DRAW SAMPLES

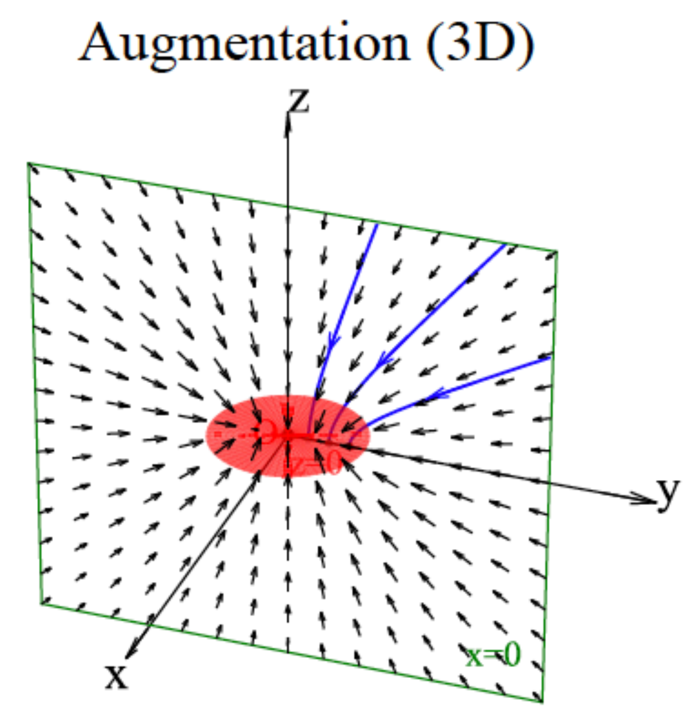
- Since $-E(x)$ points towards sources, the backward ODE $\frac{\mathrm{d}x}{\mathrm{d}t} = -E(x)$ will take samples close to the sources.
- But, the backward ODE has the problem of mode collapse
- 2D plane: points towards the center of the disk O

No augmentation (2D)



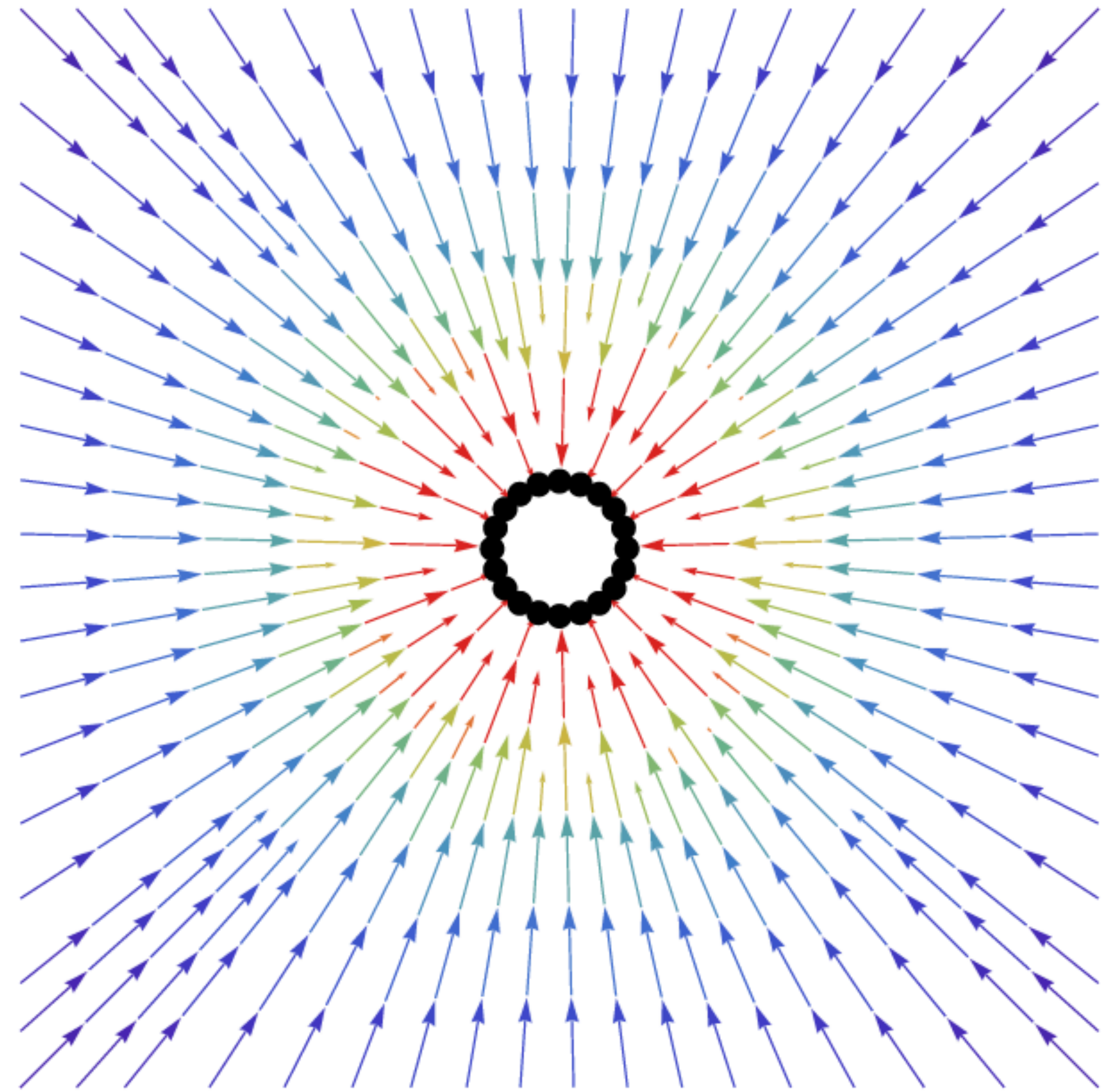
METHOD - HOW TO DRAW SAMPLES

- Since $-E(x)$ points towards sources, the backward ODE $\frac{\mathrm{d}x}{\mathrm{d}t} = -E(x)$ will take samples close to the sources.
- But, the backward ODE has the problem of mode collapse
- 2D plane: points towards the center of the disk O
- Add an additional dimension z: particles can hit different points on the disk and faithfully recover the data distribution.



METHOD - HOW TO DRAW SAMPLES

The gravitational field in which the source of gravity is isotropic



https://spaces.ac.cn/archives/9305

Thanks to Lilang Lin for introducing this blog

METHOD - HOW TO DRAW SAMPLES

- Solve the Poisson equation in an augmented space

$$\tilde{\mathbf{x}} = (\mathbf{x}, z) \in \mathbb{R}^{N+1}$$

For training data: $z = 0$

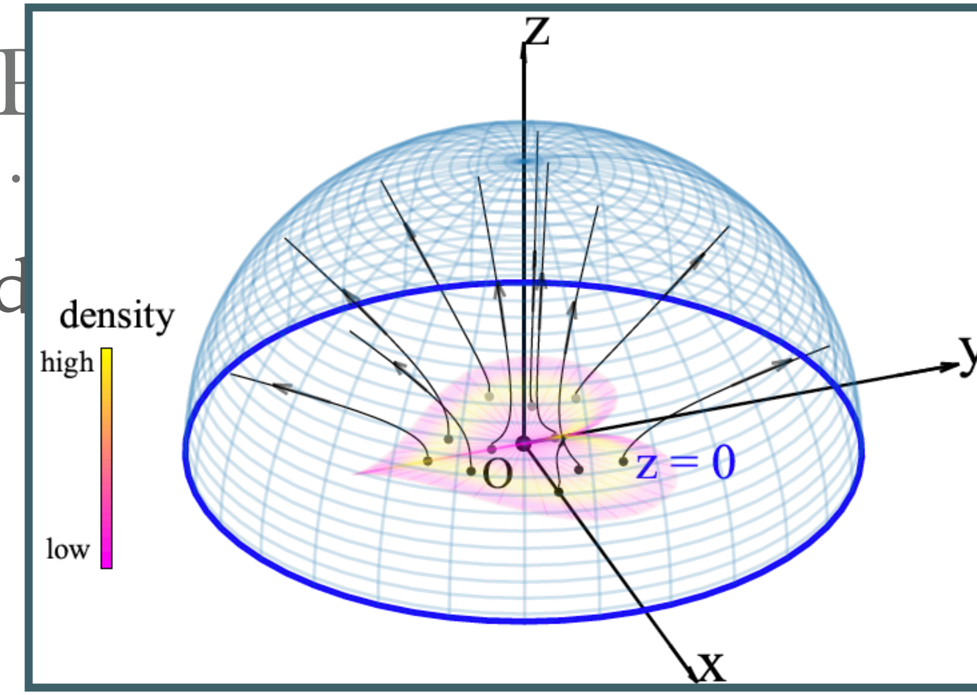
- With z, the Poisson field becomes:

$$\forall \tilde{\mathbf{x}} \in \mathbb{R}^{N+1}, \mathbf{E}(\tilde{\mathbf{x}}) = -\nabla \varphi(\tilde{\mathbf{x}}) = \frac{1}{S_N(1)} \int \frac{\tilde{\mathbf{x}} - \tilde{\mathbf{y}}}{||\tilde{\mathbf{x}} - \tilde{\mathbf{y}}||^{N+1}} \tilde{p}(\tilde{\mathbf{y}}) d\tilde{\mathbf{y}}$$

- The associated forward/backward ODEs:

$$d\tilde{\mathbf{x}}/dt = \mathbf{E}(\tilde{\mathbf{x}}), d\tilde{\mathbf{x}}/dt = -\mathbf{E}(\tilde{\mathbf{x}})$$

METHOD - HOW TO DRAW SAMP



- Solve the Poisson equation in an augmented

$$\tilde{\mathbf{x}} = (\mathbf{x}, z) \in \mathbb{R}^{N+1}$$

- For training data: $z = 0$
- With z, the Poisson field

$$\forall \tilde{\mathbf{x}} \in \mathbb{R}^{N+1}, \mathbf{E}(\tilde{\mathbf{x}}) = -\nabla$$

Define trajectories of particles between the z = 0 hyperplane and an enclosing hemisphere

- The associated forward/backward ODEs:

$$d\tilde{\mathbf{x}}/dt = \mathbf{E}(\tilde{\mathbf{x}}), d\tilde{\mathbf{x}}/dt = -\mathbf{E}(\tilde{\mathbf{x}})$$

METHOD - HOW TO DRAW SAMPLES

- Backward ODE defines a transformation between the uniform distribution on an infinite hemisphere and the data distribution $\tilde{p}(\tilde{x})$ in the z=0 plane

Theorem 1. *Suppose particles are sampled from a uniform distribution on the upper ($z > 0$) half of the sphere of radius r and evolved by the backward ODE $\frac{d\tilde{\mathbf{x}}}{dt} = -\mathbf{E}(\tilde{\mathbf{x}})$ until they reach the $z = 0$ hyperplane, where the Poisson field $\mathbf{E}(\tilde{\mathbf{x}})$ is generated by the source $\tilde{p}(\tilde{\mathbf{x}})$. In the $r \to \infty$ limit, under some mild conditions detailed in Appendix A, this process generates a particle distribution $\tilde{p}(\tilde{\mathbf{x}})$, i.e., a distribution $p(\mathbf{x})$ in the $z = 0$ hyperplane.*

- Starting from an infinite hemisphere, one can recover the data distribution $\tilde{p}$ by following the inverse Poisson field $-\mathbf{E}(x)$

METHOD - LEARNING

- With z, the Poisson field becomes:

$$\forall \tilde{\mathbf{x}} \in \mathbb{R}^{N+1}, \mathbf{E}(\tilde{\mathbf{x}}) = -\nabla\varphi(\tilde{\mathbf{x}}) = \frac{1}{S_N(1)} \int \frac{\tilde{\mathbf{x}} - \tilde{\mathbf{y}}}{||\tilde{\mathbf{x}} - \tilde{\mathbf{y}}||^{N+1}} \tilde{p}(\tilde{\mathbf{y}}) d\tilde{\mathbf{y}}$$

- Given a set of training data $D = \{x_i\}_{i=1}^{n}$ i.i.d sample from $p(x)$
- Define the empirical version of the Poisson field:

$$\hat{\mathbf{E}}(\tilde{\mathbf{x}}) = c(\tilde{\mathbf{x}}) \sum_{i=1}^{n} \frac{\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_i}{||\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_i||^{N+1}}$$

$$c(\tilde{\mathbf{x}}) = 1/ \sum_{i=1}^{n} \frac{1}{||\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_i||^{N+1}}$$

METHOD - LEARNING

- Sample points inside the hemisphere by perturbing the augmented training data

$$\mathbf{y} = \mathbf{x} + \| \epsilon_{\mathbf{x}} \| (1+\tau)^m \mathbf{u}, \quad z = |\epsilon_z|(1+\tau)^m$$

where $\epsilon = (\epsilon_{\mathbf{x}}, \epsilon_z) \sim \mathcal{N}(0, \sigma^2 I_{N+1 \times N+1})$, $\mathbf{u} \sim \mathcal{U}(S_N(1))$, $m \sim \mathcal{U}[0, M]$

M, σ, and τ are hyper-parameters

- With fixed ϵ and u, the added noise increases exponentially with m.

→ Points farther away from the data support play a less important role in generative modeling

METHOD - LEARNING

- Further normalize the field to resolve the variations

$$\mathbf{v}(\tilde{\mathbf{x}}) = -\sqrt{N}\hat{\mathbf{E}}(\tilde{\mathbf{x}})/\| \hat{\mathbf{E}}(\tilde{\mathbf{x}}) \|_2$$

  Trajectories of its forward/backward ODEs are invariant under normalization

- Mini-batch data: $\mathcal{B} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{B}|}$
- Uniformly sample *m* in [0, M] for each data
- M is large (around 300) to ensure reaching a large enough hemisphere
- Training Loss:

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \| f_\theta(\tilde{\mathbf{y}}_i) - \mathbf{v}_{\mathcal{B}_L}(\tilde{\mathbf{y}}_i) \|_2^2$$

METHOD - LEARNING

- Further normalize the field to resolve the variations

$$\mathbf{v}(\tilde{\mathbf{x}}) = -\sqrt{N}\hat{\mathbf{E}}(\tilde{\mathbf{x}})/\| \hat{\mathbf{E}}(\tilde{\mathbf{x}}) \|_2$$

  Trajectories of its forward/backward ODEs are invariant under normalization

- Mini-batch data: $\mathcal{B} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{B}|}$
- Uniformly sample m i
- M is large (around 300
- Training Loss:

Use a larger batch for the estimation of normalized field since the empirical normalized field is biased

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} \| f_\theta(\tilde{\mathbf{y}}_i) - \mathbf{v}_{\mathcal{B}_L}(\tilde{\mathbf{y}}_i) \|_2^2$$

METHOD - LEARNING

Algorithm 1 Learning the normalized Poisson Field

Input: Training iteration T, Initial model f_θ, dataset $\mathcal{D}$, constant γ, learning rate η.
for $t = 1 \dots T$ **do**
  Sample a large batch $\mathcal{B}_L$ from $\mathcal{D}$ and subsample a batch of datapoints $\mathcal{B} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{B}|}$ from $\mathcal{B}_L$
  Simulate the ODE: $\{\tilde{\mathbf{y}}_i = \text{perturb}(\mathbf{x}_i)\}_{i=1}^{|\mathcal{B}|}$
  Calculate the normalized field by $\mathcal{B}_L$: $\mathbf{v}_{\mathcal{B}_L}(\tilde{\mathbf{y}}_i) = -\sqrt{N}\hat{\mathbf{E}}_{\mathcal{B}_L}(\tilde{\mathbf{y}}_i)/(\| \hat{\mathbf{E}}_{\mathcal{B}_L}(\tilde{\mathbf{y}}_i) \|_2 + \gamma), \forall i$
  Calculate the loss: $\mathcal{L}(\theta) = \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} \| f_\theta(\tilde{\mathbf{y}}_i) - \mathbf{v}_{\mathcal{B}_L}(\tilde{\mathbf{y}}_i) \|_2^2$
  Update the model parameter: $\theta = \theta - \eta\nabla\mathcal{L}(\theta)$
end for
return f_θ

Algorithm 2 perturb($\mathbf{x}$)

Sample the power $m \sim \mathcal{U}[0, M]$
Sample the initial noise $(\epsilon_{\mathbf{x}}, \epsilon_z) \sim \mathcal{N}(0, \sigma^2 I_{(N+1)\times(N+1)})$
Uniformly sample the vector from the unit ball $\mathbf{u} \sim \mathcal{U}(S_N(1))$
Construct training point $\mathbf{y} = \mathbf{x} + \| \epsilon_{\mathbf{x}} \| (1+\tau)^m \mathbf{u}$, $z = |\epsilon_z|(1+\tau)^m$
return $\tilde{\mathbf{y}} = (\mathbf{y}, z)$

METHOD - EQUIVALENT BACKWARD ODE

➤ An equivalent backward ODE that allows for exponentially decay on z

- Sample from the data distribution by the backward ODE

$$d\tilde{\mathbf{x}} = -\mathbf{v}(\tilde{\mathbf{x}})dt$$

- The boundary condition of the above ODE is unclear:

  The starting and terminal time t are both unknown

METHOD - EQUIVALENT BACKWARD ODE

- An equivalent backward ODE

$$d(\mathbf{x}, z) = -(\frac{d\mathbf{x}}{dt}\frac{dt}{dz}dz, dz) = -(\mathbf{v}(\tilde{\mathbf{x}})_{\mathbf{x}}\mathbf{v}(\tilde{\mathbf{x}})_z^{-1}, 1)dz$$

$\mathbf{v}(\tilde{\mathbf{x}})_{\mathbf{x}}, \mathbf{v}(\tilde{\mathbf{x}})_z$ are the corresponding components of $\mathbf{x}, z$ in vector $\mathbf{v}(\tilde{\mathbf{x}})$

- When $z = 0$, we arrive at the data distribution
- We can freely choose a large z_{max} as the starting point

METHOD - EQUIVALENT BACKWARD ODE

- The distribution on the $z = z_{max}$ hyperplane is no longer uniform
- We derive the prior distribution by radially projecting uniform distribution on the hemisphere with radius $r = z_{max}$ to the $z = z_{max}$ hyperplane:

$$p_{\text{prior}}(\mathbf{x}) = \frac{2z_{\max}^{N+1}}{S_N(z_{\max})(\|\mathbf{x}\|_2^2 + z_{\max}^2)^{\frac{N+1}{2}}} = \frac{2z_{\max}}{S_N(1)(\|\mathbf{x}\|_2^2 + z_{\max}^2)^{\frac{N+1}{2}}}$$

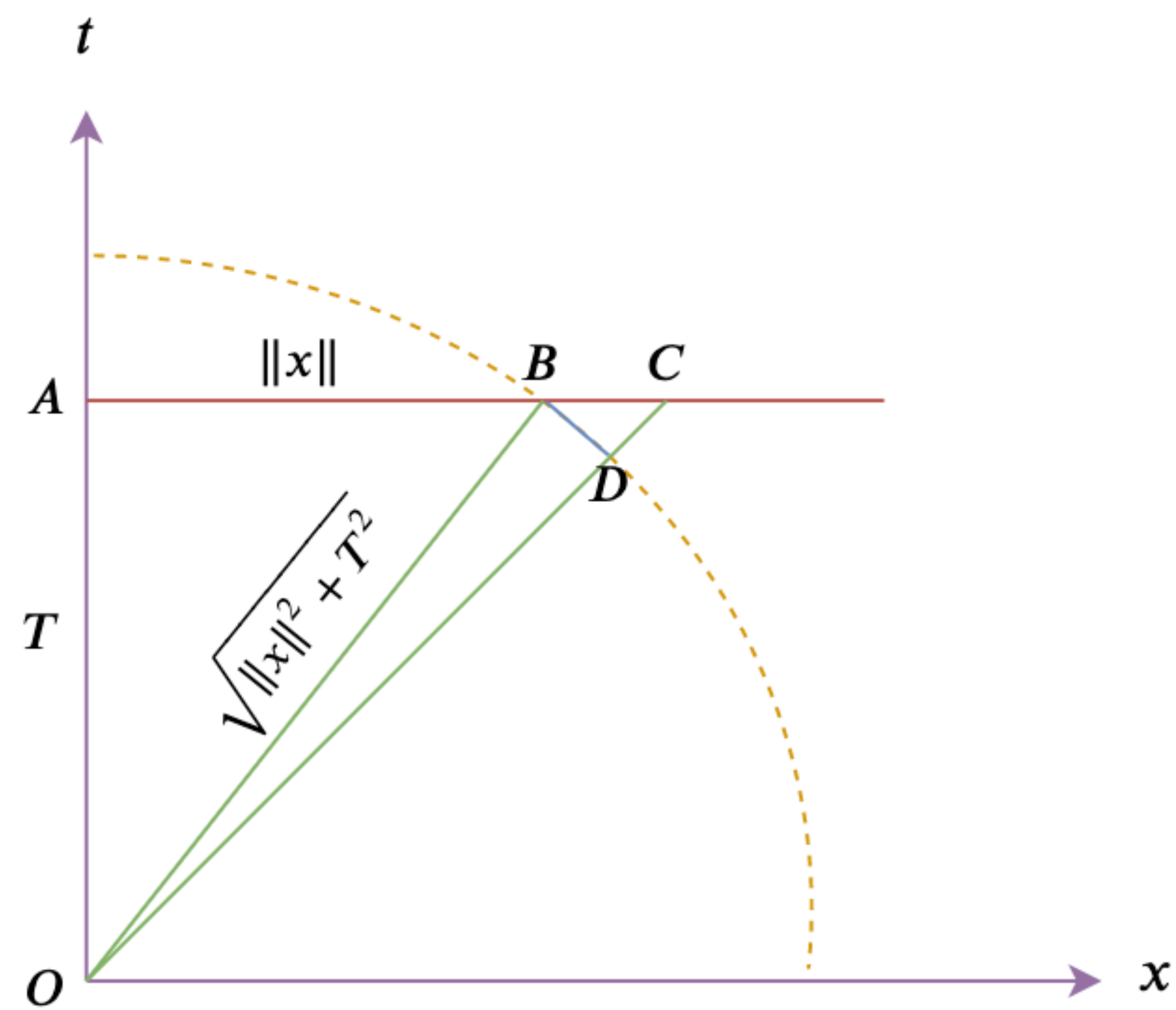
where $S_N(r)$ is the surface area of N-sphere with radius r.

METHOD - EQUIVALENT BACKWARD ODE

Projection of the sphere onto the t=T plane

When B and D are very close

$$\frac{|BC|}{|BD|} = \frac{|OB|}{|OA|} = \frac{\sqrt{\|\boldsymbol{x}\|^2 + T^2}}{T}$$



https://spaces.ac.cn/archives/9305

Thanks to Lilang Lin for introducing this blog

OUTLINE

EXPERIMENTS

- CIFAR-10 sample quality (FID, Inception) and number of function evaluation (NFE).

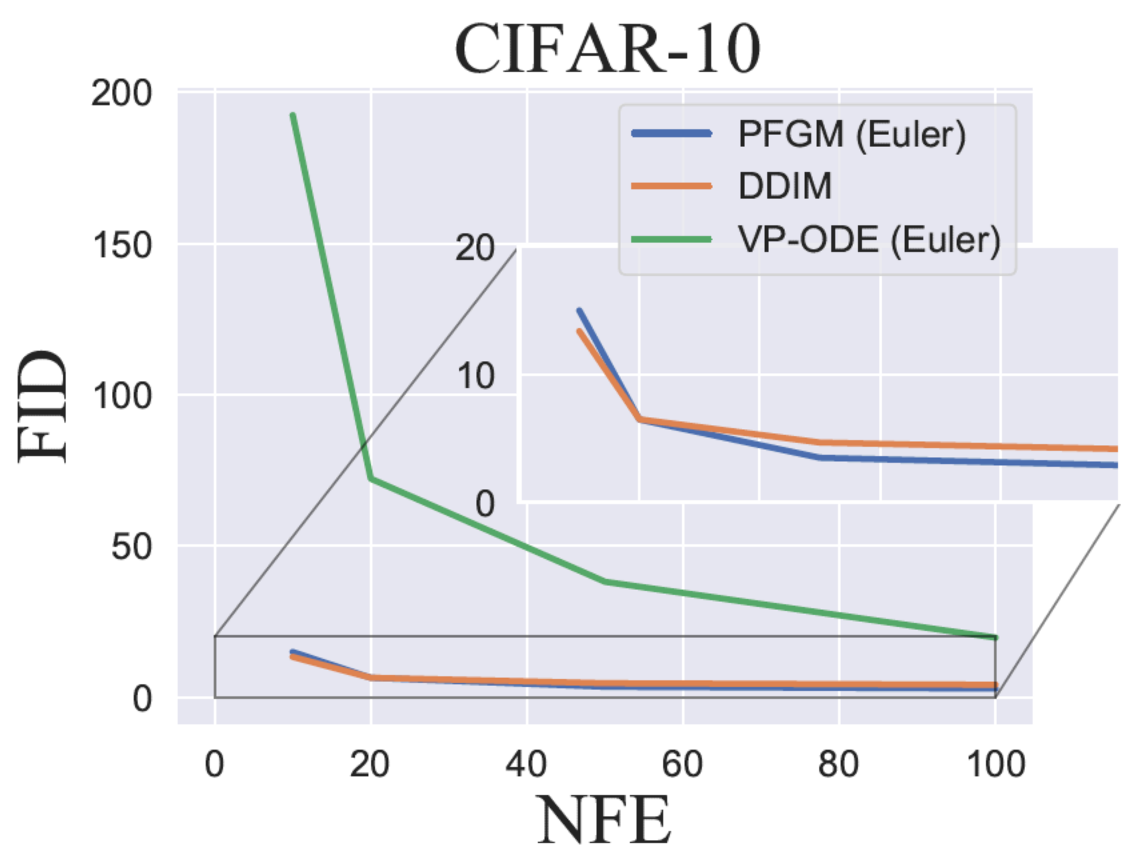
	Invertible?	Inception ↑	FID ↓	NFE ↓
PixelCNN [36]	✗	4.60	65.9	1024
IGEBM [8]	✗	6.02	40.6	60
ViTGAN [24]	✗	9.30	6.66	1
StyleGAN2-ADA [17]	✗	9.83	2.92	1
StyleGAN2-ADA (cond.) [17]	✗	10.14	2.42	1
NCSN [31]	✗	8.87	25.32	1001
NCSNv2 [32]	✗	8.40	10.87	1161
DDPM [16]	✗	9.46	3.17	1000
NCSN++ VE-SDE [33]	✗	9.83	2.38	2000
NCSN++ deep VE-SDE [33]	✗	9.89	2.20	2000
Glow [19]	✓	3.92	48.9	1
DDIM, T=50 [30]	✓	-	4.67	50
DDIM, T=100 [30]	✓	-	4.16	100
NCSN++ VE-ODE [33]	✓	9.34	5.29	194
NCSN++ deep VE-ODE [33]	✓	9.17	7.66	194
DDPM++ backbone				
VP-SDE [33]	✗	9.58	2.55	1000
sub-VP-SDE [33]	✗	9.56	2.61	1000
VP-ODE [33]	✓	9.46	2.97	134
sub-VP-ODE [33]	✓	9.30	3.16	146
PFGM (ours)	✓	**9.65**	**2.48**	**104**
DDPM++ deep backbone				
VP-SDE [33]	✗	9.68	2.41	1000
sub-VP-SDE [33]	✗	9.57	2.41	1000
VP-ODE [33]	✓	9.47	2.86	134
sub-VP-ODE [33]	✓	9.40	3.05	146
PFGM (ours)	✓	**9.68**	**2.35**	**110**

EXPERIMENTS

Figure 3: Uncurated samples on datasets of increasing resolution. From left to right: CIFAR-10 32×32, CelebA 64×64 and LSUN bedroom 256×256.

EXPERIMENTS

- To accelerate the inference speed of ODEs, we can increase the step size (decrease the NFEs) in numerical solvers



(c) FID vs. NFE on CIFAR-10

The effects of increasing step size on PFGM, VP-ODE and DDIM using the forward Euler method, with a varying NFE ranging from 10 to 100.

EXPERIMENTS

- Define the invertible forward mapping:

$$\mathbf{x}(\log z_{\max}) = \mathcal{M}(\mathbf{x}(\log z_{\min})) \equiv \mathbf{x}(\log z_{\min}) + \int_{\log z_{\min}}^{\log z_{\max}} \mathbf{v}(\mathbf{x}(t'))_{\mathbf{x}} \mathbf{v}(\tilde{\mathbf{x}}(t'))_z^{-1} e^{t'} dt'$$

- Likelihood evaluation

Table 2: Bits/dim on CIFAR-10

	bits/dim ↓
RealNVP [6]	3.49
Glow [19]	3.35
Residual Flow [3]	3.28
Flow++ [14]	3.29
DDPM (L) [16]	$\leq 3.70^*$
DDPM++ backbone	
VP-ODE [33]	3.20
sub-VP-ODE [33]	**3.02**
PFGM (ours)	3.19

EXPERIMENTS

- Image Interpolations

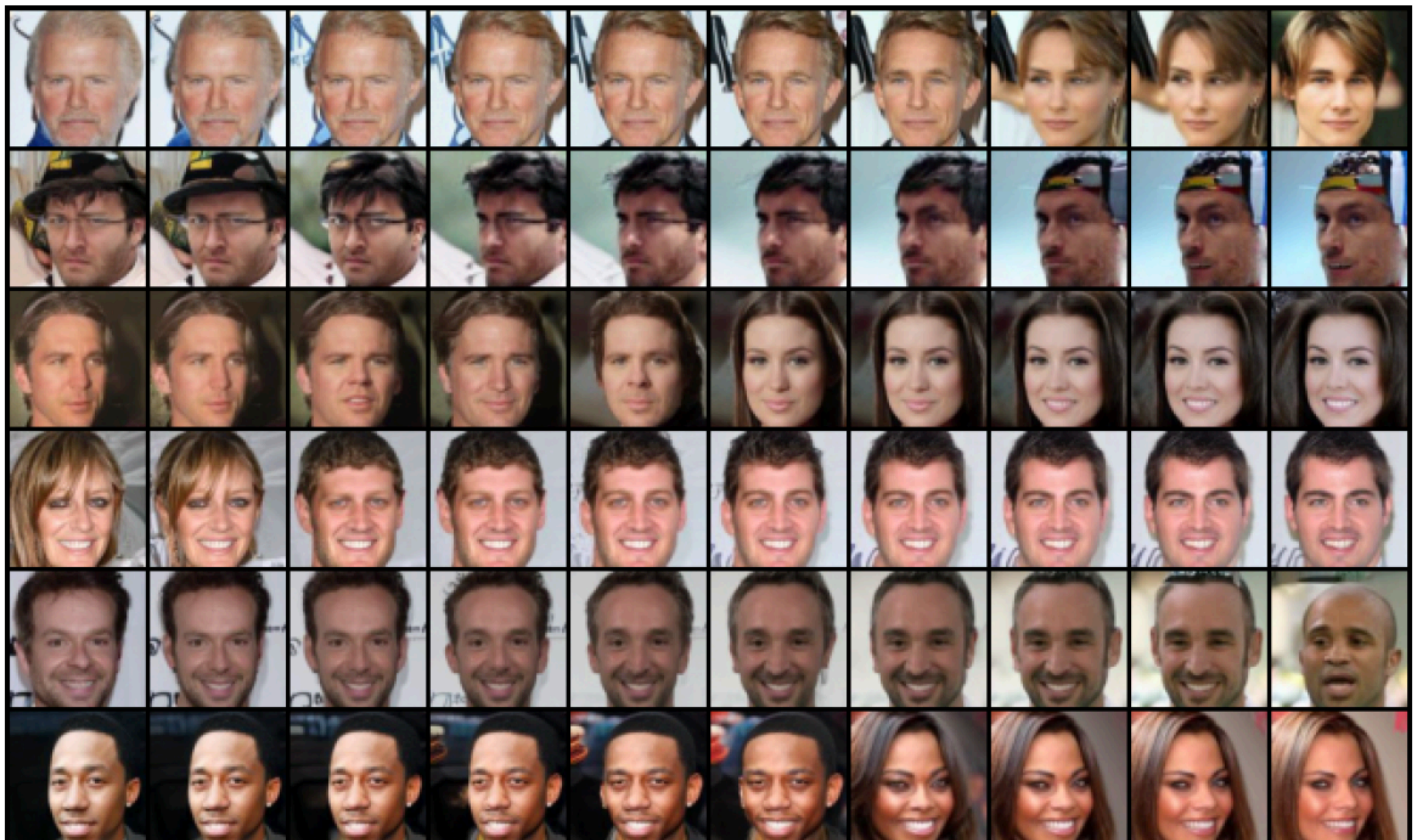


Figure 10: Interpolation on CelebA 64×64 by PFGM

OUTLINE

CONCLUSION

- “Poisson flow” generative model (PFGM)
- Interpret the data points as electrical charges on the $z = 0$ hyperplane in a space augmented with an additional dimension z