Nonparametric Generative Modeling with Conditional Sliced-Wasserstein Flows

Chao Du, Tianbo Li, Tianyu Pang, Shuicheng Yan, Min Lin

Sea AI Lab, Singapore

ICML 2023

Overview

Topic: Nonparametric method & Conditional generative modeling

Overview

Topic: Nonparametric method & Conditional generative modeling

Contribution:

- Reveal the conditional modeling capabilities of Sliced-Wasserstein Flows
- Introduce inductive biases for image tasks into Sliced-Wasserstein Flows

Overview

Topic: Nonparametric method & Conditional generative modeling

Contribution:

- Reveal the <u>conditional modeling capabilities</u> of Sliced-Wasserstein Flows
- Introduce <u>inductive biases for image tasks</u> into Sliced-Wasserstein Flows

Takeaways:

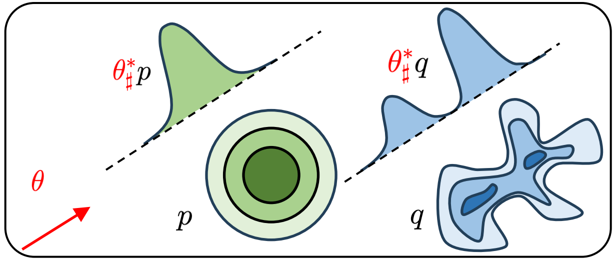
- The first nonparametric conditional generative model (to our best knowledge)
- Achieve comparable performance with parametric generative models

Sliced-Wasserstein Flows (SWF)

Sliced-Wasserstein distance:

Based on projections: $\theta^*(x) \triangleq \langle \theta, x \rangle$

$$\boxed{SW_2^2(p,q) \triangleq \int_{\mathbb{S}^{d-1}} W_2^2(\theta^*_\sharp p, \theta^*_\sharp q)\mathrm{d}\theta}$$

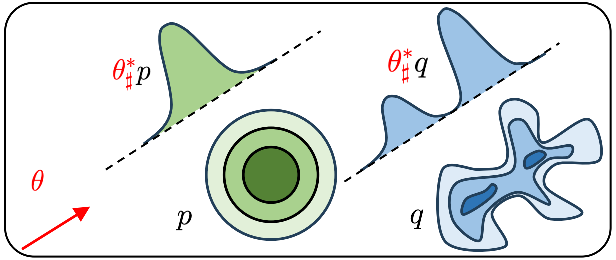


Sliced-Wasserstein Flows (SWF)

Sliced-Wasserstein distance:

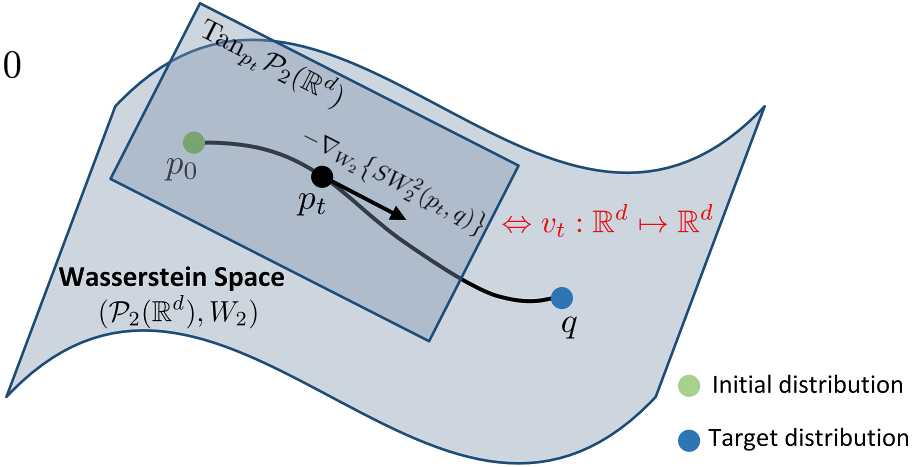
Based on projections: $\theta^*(x) \triangleq \langle \theta, x \rangle$

$$\boxed{SW_2^2(p,q) \triangleq \int_{\mathbb{S}^{d-1}} W_2^2(\theta^*_\sharp p, \theta^*_\sharp q) \mathrm{d}\theta}$$



SWF: Gradient flow in the Wasserstein space minimizing SW_2

$$\min_{p \in \mathcal{P}_2(\mathbb{R}^d)} SW_2^2(p,q) \Rightarrow \frac{\partial p_t(x)}{\partial t} + \nabla \cdot (p_t(x) v_t(x)) = 0$$

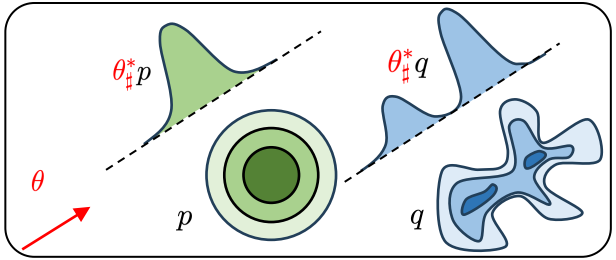


Sliced-Wasserstein Flows (SWF)

Sliced-Wasserstein distance:

Based on projections: $\theta^*(x) \triangleq \langle \theta, x \rangle$

$$SW_2^2(p,q) \triangleq \int_{\mathbb{S}^{d-1}} W_2^2(\theta^*_\sharp p, \theta^*_\sharp q) \mathrm{d}\theta$$



SWF: Gradient flow in the Wasserstein space minimizing SW_2

$$\min_{p \in \mathcal{P}_2(\mathbb{R}^d)} SW_2^2(p,q) \quad \Rightarrow \quad \frac{\partial p_t(x)}{\partial t} + \nabla \cdot (p_t(x) v_t(x)) = 0$$

$$v_t(x) \triangleq - \boxed{\int_{\mathbb{S}^{d-1}}} \psi'_{t,\theta}(\theta^\top x) \cdot \theta \; \mathrm{d}\theta$$

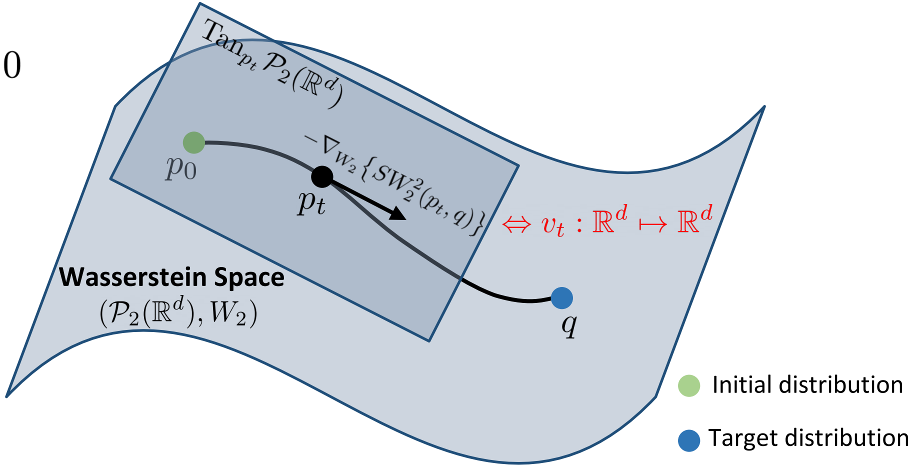
Monte Carlo estimation over unit sphere

Velocity field:

Bonnotte 2013; Liutkus et al., 2019

$$\psi'_{t,\theta}(z) = z - \boxed{F^{-1}_{\theta^*_\sharp q} \circ F_{\theta^*_\sharp p_t}}(z)$$

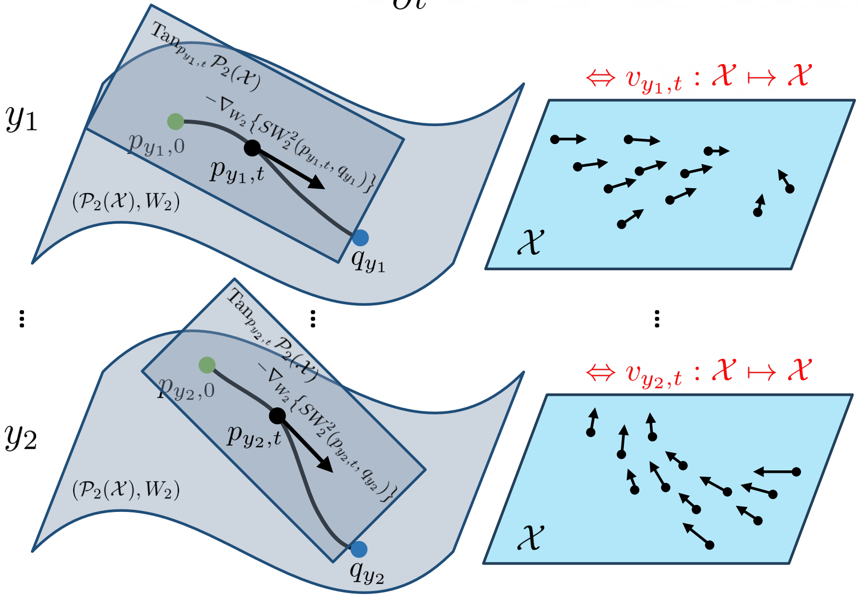
one-dimensional CDF estimations



Conditional SWF

Ideally: Collection of SWFs $\min_{p_y \in \mathcal{P}_2(\mathcal{X})} SW_2^2(p_y, q_y), \forall y \in \mathcal{Y}$

$\Rightarrow$ $(p_{y,t})_{t\geq 0}$ solves $\dfrac{\partial p_{y,t}(x)}{\partial t} + \nabla \cdot (p_{y,t}(x) v_{y,t}(x)) = 0$

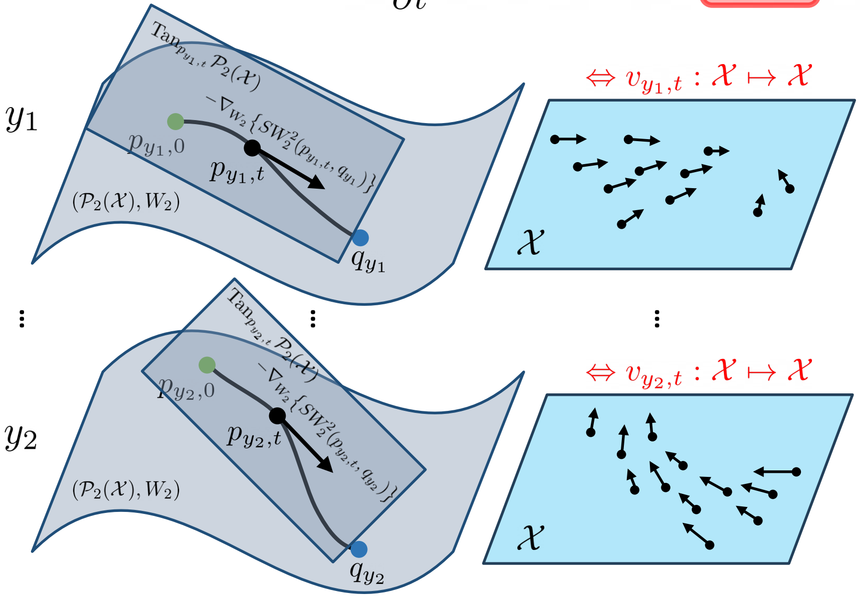


data efficiency ☒ generalization ☒

Conditional SWF

Ideally: Collection of SWFs $\min_{p_y \in \mathcal{P}_2(\mathcal{X})} SW_2^2(p_y, q_y), \forall y \in \mathcal{Y}$

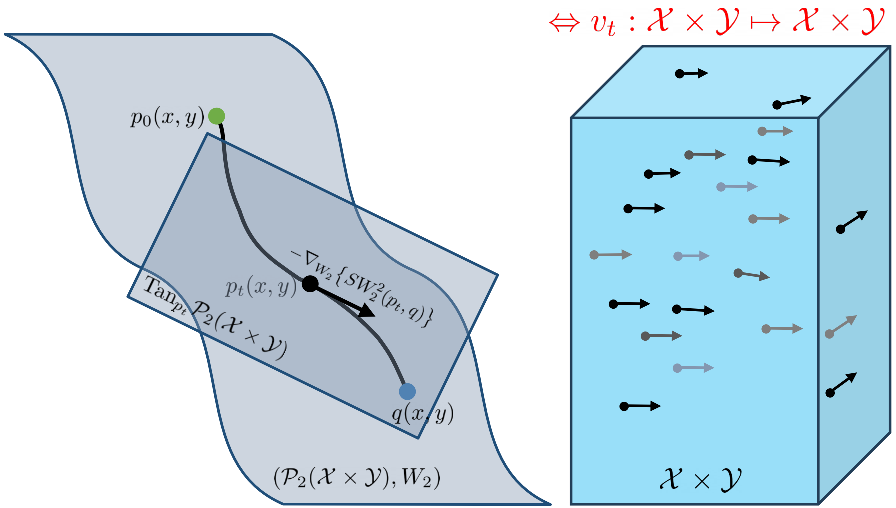
$\Rightarrow (p_{y,t})_{t\geq 0}$ solves $\frac{\partial p_{y,t}(x)}{\partial t} + \nabla \cdot (p_{y,t}(x) \boxed{v_{y,t}(x)}) = 0$



data efficiency $\boxtimes$ generalization $\boxtimes$

This work: SWF in the joint space $\min_{p \in \mathcal{P}_2(\mathcal{X}\times\mathcal{Y})} SW_2^2(p, q)$

$\Rightarrow (p_t)_{t\geq 0}$ solves $\frac{\partial p_t(x,y)}{\partial t} + \nabla \cdot (p_t(x,y) \boxed{v_t(x,y)}) = 0$

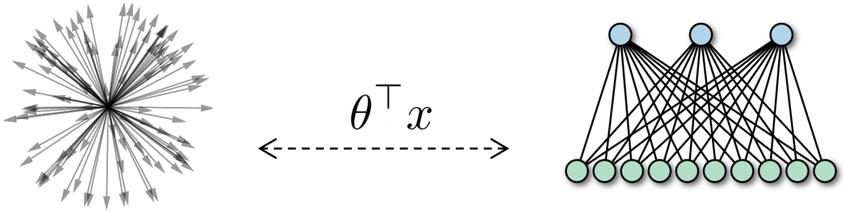


Observation: (under certain conditions) the velocities coincide!

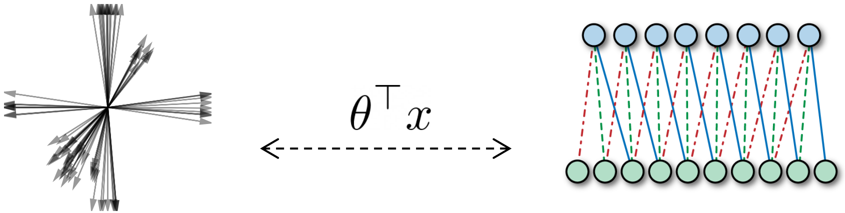
$$\boxed{v_t^{\mathcal{X}}(x,y)} \approx \boxed{v_{y,t}(x)}$$
$$\boxed{v_t^{\mathcal{Y}}(x,y)} \approx 0$$

Introducing Inductive Biases

Uniform projections ⇔ Fully-connected layers

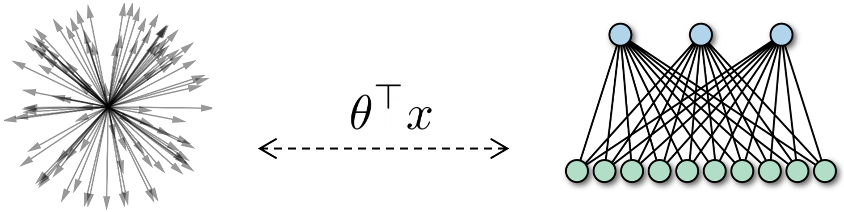


Locally-connected projections ⇔ Locally-connected layers

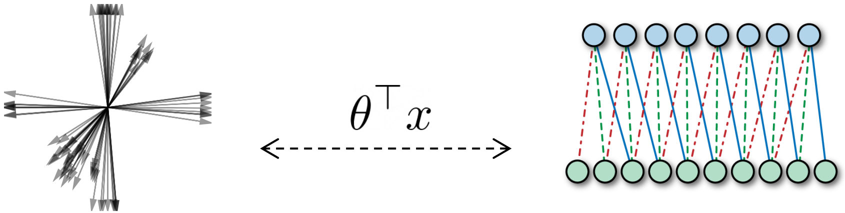


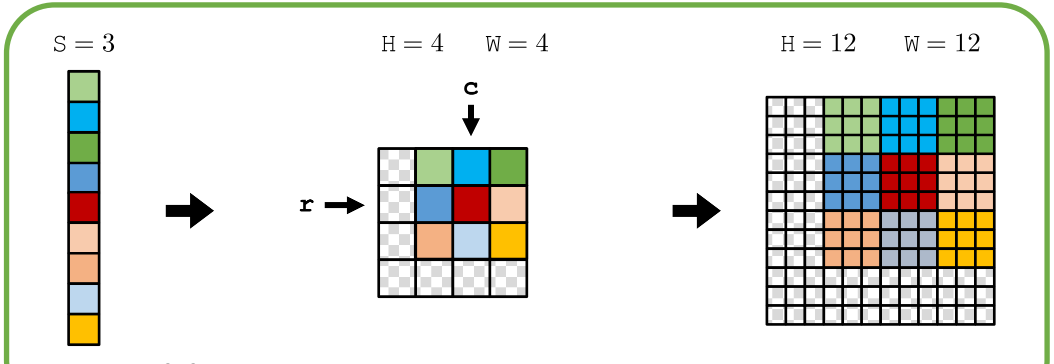
Introducing Inductive Biases

Uniform projections ⇔ Fully-connected layers



Locally-connected projections ⇔ Locally-connected layers





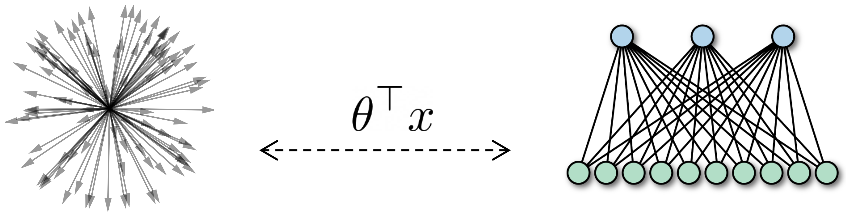
$\theta_{\text{patch}} \in \mathbb{R}^{3\times 3}$

Locally-connected projection

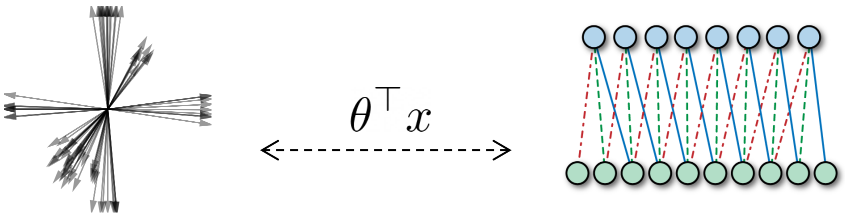
Upsampled projection

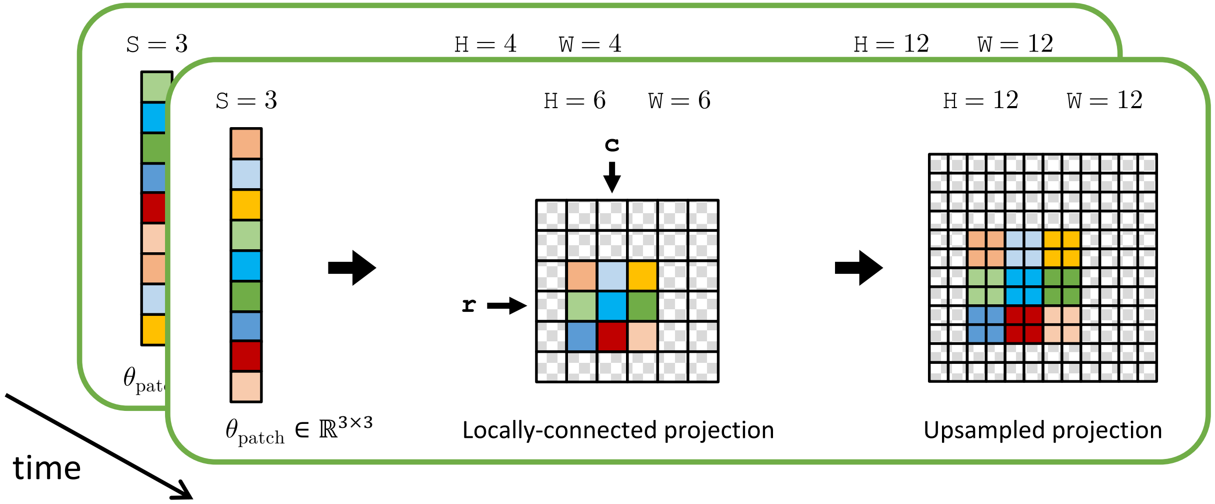
Introducing Inductive Biases

Uniform projections ⇔ Fully-connected layers



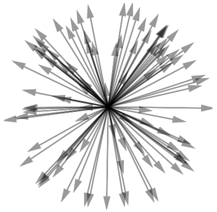
Locally-connected projections ⇔ Locally-connected layers



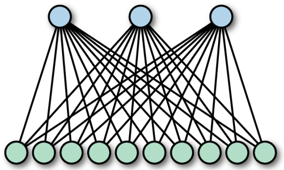


Introducing Inductive Biases

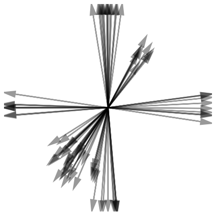
Uniform projections ⇔ Fully-connected layers



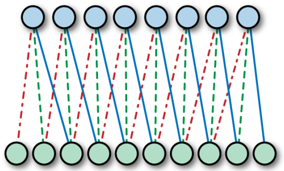
$$\theta^\top x$$



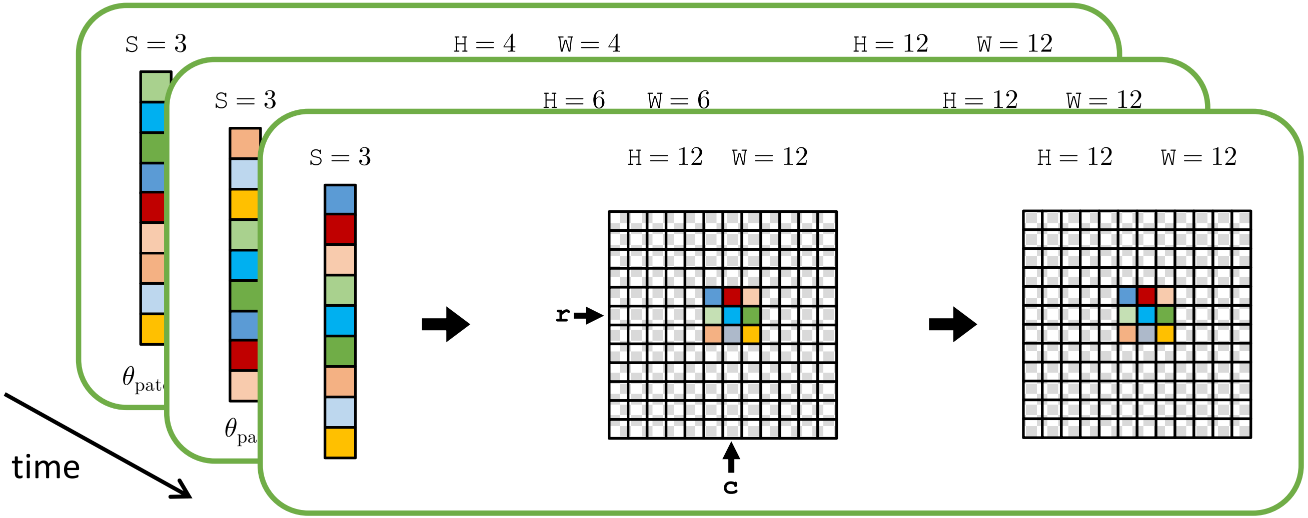
Locally-connected projections ⇔ Locally-connected layers

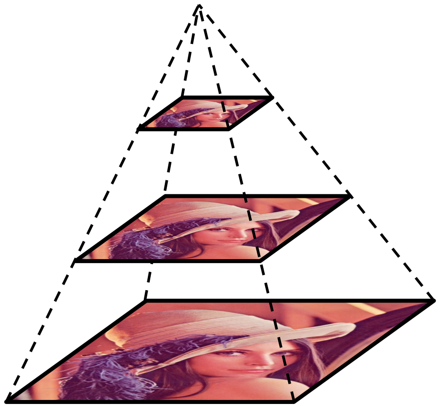


$$\theta^\top x$$



Pyramidal Schedules ⇔ Pyramidal Representation





Locally-Connected Projections & Pyramidal Schedules

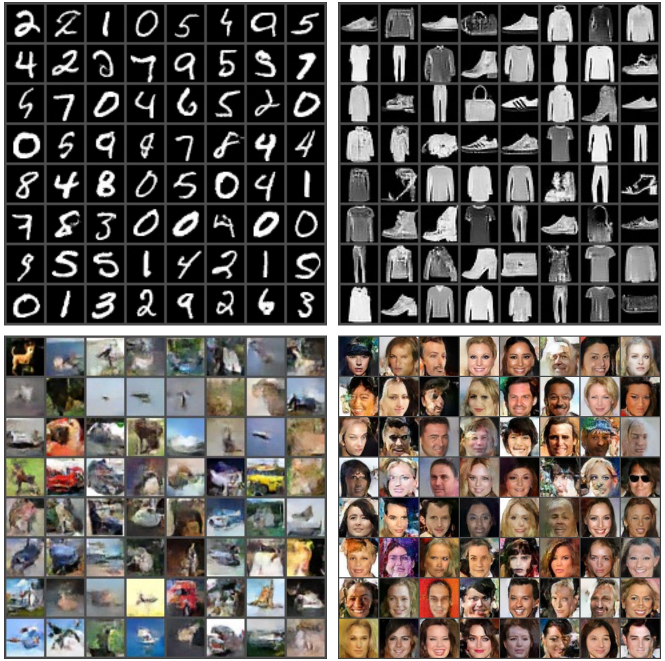
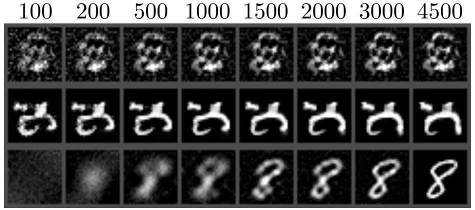
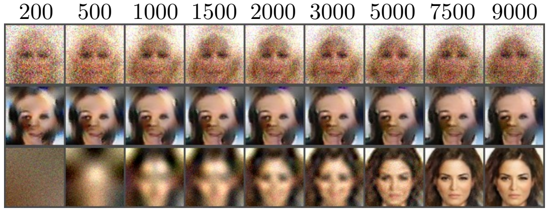


Table 1. FID↓ scores obtained by ℓ-SWF on CIFAR-10 and CelebA. ⋄ Use 160 × 160 center-cropping. ∗ Use 128 × 128 center-cropping. † Use 140 × 140 center-cropping.

Method	CIFAR-10	CelebA
Auto-encoder based		
VAE (Kingma & Welling, 2013)	155.7	$85.7^{\diamond}$
SWAE (Wu et al., 2019)	107.9	48.9^{*}
WAE (Tolstikhin et al., 2017)	–	$42^{\dagger}$
CWAE (Knop et al., 2020)	120.0	$49.7^{\dagger}$
Autoregressive & Energy based		
PixelCNN (Van den Oord et al., 2016)	65.9	–
EBM (Du & Mordatch, 2019)	37.9	–
Adversarial		
WGAN (Arjovsky et al., 2017)	55.2	$41.3^{\diamond}$
WGAN-GP (Gulrajani et al., 2017)	55.8	$30.0^{\diamond}$
CSW (Nguyen & Ho, 2022b)	36.8	–
SWGAN (Wu et al., 2019)	17.0	13.2^{*}
Score based		
NCSN (Song & Ermon, 2019)	25.3	–
Nonparametric		
SWF (Liutkus et al., 2019)	> 200	$> 150^{\dagger}$
SINF (Dai & Seljak, 2021)	66.5	37.3^{*}
ℓ-SWF (Ours)	59.7	$38.3^{\dagger}$





SWF (Liutkus et al., 2019)

SWF + Locally-Connected Projections

SWF + Locally-Connected Projections + Pyramidal Schedules

Conditional Generation

Class-conditional samples from CSWF on MNIST, Fashion MNIST and CIFAR-10.

Image Inpainting

Image inpainting results of CSWF on MNIST, Fashion MNIST, CIFAR-10 and CelebA.

For more technical details and results, please visit

Poster:

Exhibit Hall 1 #120 (Poster Session 1, 11:00 AM to 1:30 PM on July 25)

Code:

https://github.com/duchao0726/Conditionial-SWF

sea connecting the dots | AI Lab

40 Years ICML International Conference On Machine Learning