



**HAL**  
open science

# Unifying GANs and Score-Based Diffusion as Generative Particle Models

Jean-Yves Franceschi, Mike Gartrell, Ludovic Dos Santos, Thibaut Issenhuth,  
Emmanuel de Bézenac, Mickaël Chen, Alain Rakotomamonjy

► **To cite this version:**

Jean-Yves Franceschi, Mike Gartrell, Ludovic Dos Santos, Thibaut Issenhuth, Emmanuel de Bézenac, et al.. Unifying GANs and Score-Based Diffusion as Generative Particle Models. Alice Oh; Tristan Naumann; Amir Globerson; Kate Saenko; Moritz Hardt; Sergey Levine. 37th Conference on Neural Information Processing Systems, Dec 2023, New Orleans, United States. Curran Associates, Inc., Advances in Neural Information Processing Systems, 36, 2023. hal-04322365

**HAL Id: hal-04322365**

**<https://hal.science/hal-04322365v1>**

Submitted on 4 Dec 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License

# Unifying GANs and Score-Based Diffusion as Generative Particle Models

## Our Contributions

- We unify gradient flows, score-based diffusion models, and GANs in a single framework.
- We represent generated data as moving particles. A model is defined by:
  - a gradient vector field that the particles follow either at inference or training time;
  - the possibility of incorporating a generator that smooths this movement.
- This suggests the existence of hybrid models:
  - a generator trained with diffusion guidance (Score GANs);
  - a GAN trained without a generator (Discriminator Flows).
- We experimentally verify our findings.

## GANs vs Diffusion

Traditional opposition in the literature.

- |  |  |
|--|--|
| <p><b>GANs</b> → Generator trained by discriminating true vs fake data.</p> <ul style="list-style-type: none"> <li>Generator (manifold learning).</li> <li>Close to SOTA performance.</li> <li>Harder to optimize.</li> <li>Fast inference.</li> </ul> | <p><b>Diffusion</b> → Learns to progressively reverse a data degradation process.</p> <ul style="list-style-type: none"> <li>No generator (on the data space).</li> <li>SOTA performance.</li> <li>Easier to optimize.</li> <li>Slow inference.</li> </ul> |
|--|--|

## From PMs to Int-PMs

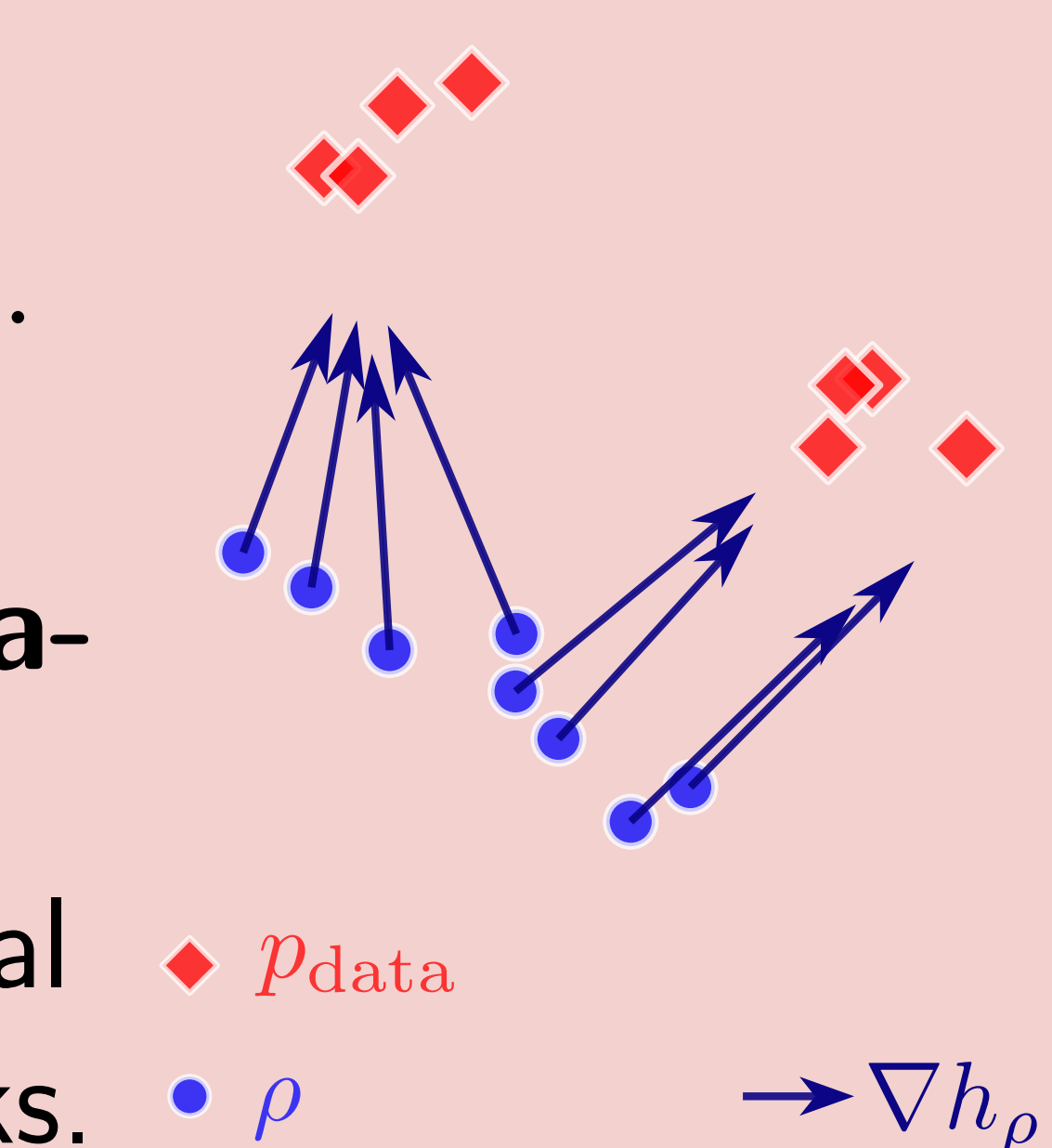
- We assign to each generated particle  $x = g_\theta(z)$  the same loss as in PMs:  $\mathcal{L}_{\text{gen}}(\theta) = -\mathbb{E}_{z \sim p_z} [h_{\rho_t}(g_\theta(z))]$ .
- We do not take into account the dependency of  $\rho_t$  w.r.t.  $\theta_t$ , to mimic PMs:  $\rho = \text{StopGradient}(g_\theta \# p_z)$ .
- Continuous-time gradient descent:
 
$$\frac{d\theta_t}{dt} = -\eta \nabla_{\theta} \mathcal{L}_{\text{gen}}(\theta_t) = \eta \nabla_{\theta} \mathbb{E}_{z \sim p_z} [h_{\rho_t}(g_\theta(z))]$$

$$= \eta \mathbb{E}_{z \sim p_z} [\nabla_{\theta} g_\theta(z) \nabla h_{\rho_t}(g_\theta(z))].$$
- Evolution of particles:
 
$$\frac{dg_\theta(z)}{dt} = \nabla_{\theta} g_\theta(z) \top \frac{d\theta_t}{dt} = \eta \mathbb{E}_{z' \sim p_z} [\nabla_{\theta} g_{\theta_t}(z) \top \nabla_{\theta} g_{\theta_t}(z') \nabla h_{\rho_t}(g_{\theta_t}(z'))].$$

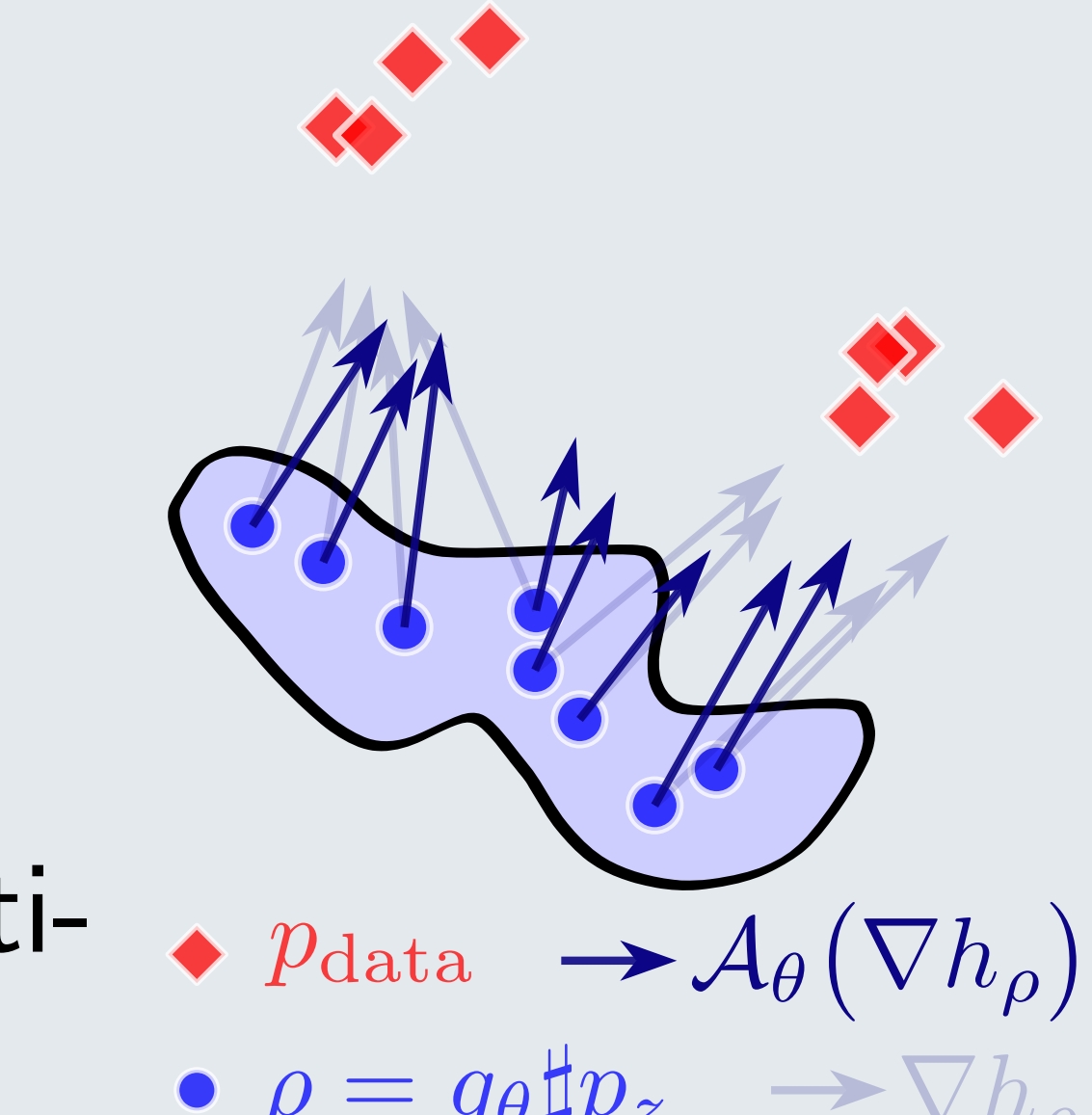
## Particle-Based Framework

Generated particles  $x_t \sim \rho_t$  follow a gradient vector field  $\nabla h_{\rho_t}$ , i.e. optimize an objective  $h_{\rho_t}$ .

### Particle Models (No Generator)

- At generation / inference time  $t$ :  $x_0 \sim \pi = \rho_0$ ,  $dx_t = \nabla h_{\rho_t}(x_t) dt$ .
  - Independently moving particles.
  - Each  $x_t$  individually follows a **gradient ascent** path on  $h_{\rho_t}(x_t)$ .
  - $h_{\rho}$  is usually a predefined functional approximated with neural networks.
- 

### Interacting Particle Models (Generator)

- Training with **the same loss**:  $\mathcal{L}_{\text{gen}}(\theta) = -\mathbb{E}_{z \sim p_z} [h_{\rho_t}(g_\theta(z))]$ .
  - At training time  $t$ :  $dg_{\theta_t}(z) = \eta [A_{\theta_t}(z)] (\nabla h_{\rho_t}) dt$ .
  - Generalization of PMs where particles interact with each other.
- 

## Smoothing Operator

- $A_{\theta}(z)$  is a linear operator on vector fields (kernel integral operator):  $[A_{\theta}(z)](V) \triangleq \mathbb{E}_{z' \sim p_z} [k_{g_{\theta_t}}(z, z') V(g_{\theta_t}(z'))]$ ,  $k_{g_{\theta_t}}(z, z') \triangleq \nabla_{\theta} g_{\theta_t}(z) \top \nabla_{\theta} g_{\theta_t}(z')$ .
- $k_{g_{\theta_t}}$  is the generator's matrix Neural Tangent Kernel (NTK, Jacot et al., 2018).
- Special case:  $k_{g_{\theta_t}}(z, z') = \delta_{z-z'} I_d$  (generator with infinite capacity).
  - No interaction between particles:  $[A_{\theta_t}(z)](V) = V(g_{\theta_t}(z))$ .
  - $dg_{\theta_t}(z) = \nabla h_{\rho_t}(g_{\theta_t}(z)) dt$ : we retrieve PMs.
- General case:  $A_{\theta_t}$  represents the parameterization of  $\rho$  as a manifold.
  - $A_{\theta_t}$  smooths the original vector field  $\nabla h_{\rho_t}$  by convolving it with  $k$ .
  - Particles interact with each other through generator parameterization.

## Wasserstein Gradient

$$-\nabla_W \mathcal{F}(\rho_t) = -\nabla \frac{\partial \mathcal{F}(\rho_t)}{\partial \rho_t}$$

### Wasserstein Gradient Flows

- Gradient descent for functionals over distributions  $\mathcal{F}$  (Santambrogio, 2017).

	Objective $\mathcal{F}(\rho)$	$h_{\rho}$
Forward KL	$\mathbb{E}_{\rho} \log \rho / p_{\text{data}}$	$-\log \rho / p_{\text{data}}$
$f$ -divergence	$\mathbb{E}_{p_{\text{data}}} f(\rho / p_{\text{data}})$	$-f'(\rho / p_{\text{data}})$
Squared MMD w.r.t. kernel $k$	$\mathbb{E}_{x, x' \sim \rho} [k(x, x')] + \mathbb{E}_{y, y' \sim p_{\text{data}}} [k(y, y')] - 2\mathbb{E}_{x \sim \rho, y \sim p_{\text{data}}} [k(x, y)]$	$\mathbb{E}_{y \sim p_{\text{data}}} [k(y, \cdot)] - \mathbb{E}_{x \sim \rho} [k(x, \cdot)]$
Entropy	$\mathbb{E}_{\rho} \log \rho$	$-\log \rho$

## Stein Gradient Flows

- Stein gradient flows (Liu, 2017) are kernelized Wasserstein gradient flows:  $dx_t = \mathbb{E}_{x_t' \sim \rho_t} [k(x_t, x_t') \nabla h_{\rho_t}(x_t')] dt$ .
- Int-PMs under mild hypotheses (generalization of Durr et al. (2022)).
- Hint towards the same  $h_{\rho}$  being used in a PM and an Int-PM.

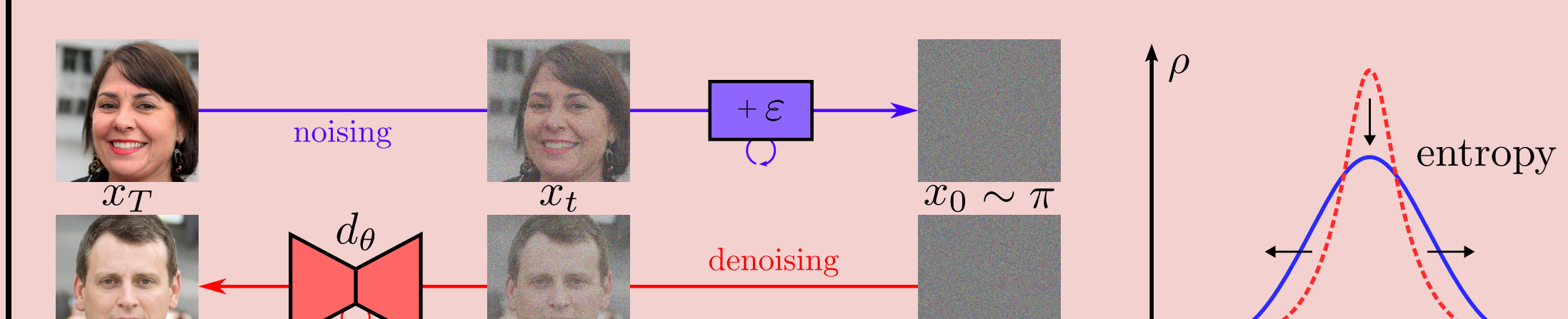
## Other Models & Flows

- Int-PMs and Stein (generalization of Durr et al. (2022)):  $k(g_{\theta_t}(z), g_{\theta_t}(z')) = k_{g_{\theta_t}}(z, z')$  in the NTK regime.
- Langevin diffusion (Song et al., 2019) is a KL flow.
- Under some hypotheses, GANs are Stein flows (Franceschi et al., 2022; Yi et al., 2023): KL flow for  $f$ -divergence GANs, squared MMD for IPM GANs.
- As a consequence, under similar hypotheses, Discriminator Flows with the same losses are Wasserstein flows.
- Many methods use neural networks to approximate the flow (Alvarez-Melis et al., 2022; Heng et al., 2023).

## Log Ratio Gradient

$$\alpha_t \nabla \log [p_{\text{data}} \star k_{\text{RBF}}^{\sigma(t)}] - \beta_t \nabla \log \rho_t$$

### Score-Based Diffusion

- Using Jordan et al. (1998) in Song et al. (2019) and Karras et al. (2022):  $dx_t = \alpha_t \nabla \log [p_{\text{data}} \star k_{\text{RBF}}^{\sigma(t)}](x_t) dt + \sqrt{2\beta_t} dW_t \Leftrightarrow -\beta_t \nabla \log \rho_t dt$ .
- 

## Score GANs

- Estimate  $\nabla h_{\rho}$  with two score matching networks and use it in parameter update equation of generator training.
- Data score (pretrained like diffusion):  $s_{\psi}^{p_{\text{data}}}(\cdot, \sigma) \equiv \nabla \log [p_{\text{data}} \star k_{\text{RBF}}^{\sigma}]$ .
- Gen. score (continuously updated like a discriminator):  $s_{\phi}^{\rho} \equiv \nabla \log \rho$ .

## Score GANs in Practice


- Two practical issues:
  - sliced score matching to train  $s_{\phi}^{\rho}$ ;
  - scheduling  $\sigma$ s w.r.t. training time  $t$ .
- We randomly sample  $\sigma$  and also noise the particles:  $\nabla h_{\rho} = \nabla \log [p_{\text{data}} \star k_{\text{RBF}}^{\sigma}] - \nabla \log [\rho_t \star k_{\text{RBF}}^{\sigma}]$ ,  $\equiv \widetilde{\nabla} h_{\rho}(\cdot, \sigma) = s_{\psi}^{p_{\text{data}}}(\cdot, \sigma) - s_{\phi}^{\rho}(\cdot, \sigma)$ .
- Generator update:
  - few-step training of  $s_{\phi}^{\rho}$  with denoising score matching;
  - gradient descent step:  $\theta \leftarrow \theta + \eta \mathbb{E}_{\sigma \sim p_{\sigma}, \varepsilon \sim \mathcal{N}(0, \sigma I_D), z \sim p_z} [\nabla_{\theta} g_{\theta}(z) \widetilde{\nabla} h_{\rho}(g_{\theta}(z) + \varepsilon, \sigma)]$ .

## Discriminator Gradient

$$-\nabla (c \circ f_{\rho_t})$$

where  $f_{\rho_t}$  discriminates  $\rho_t$  from  $p_{\text{data}}$

### Discriminator Flows

- Particles directly follow the discriminator gradient.
  - The discriminator is simultaneously trained and used to generate data.
- 

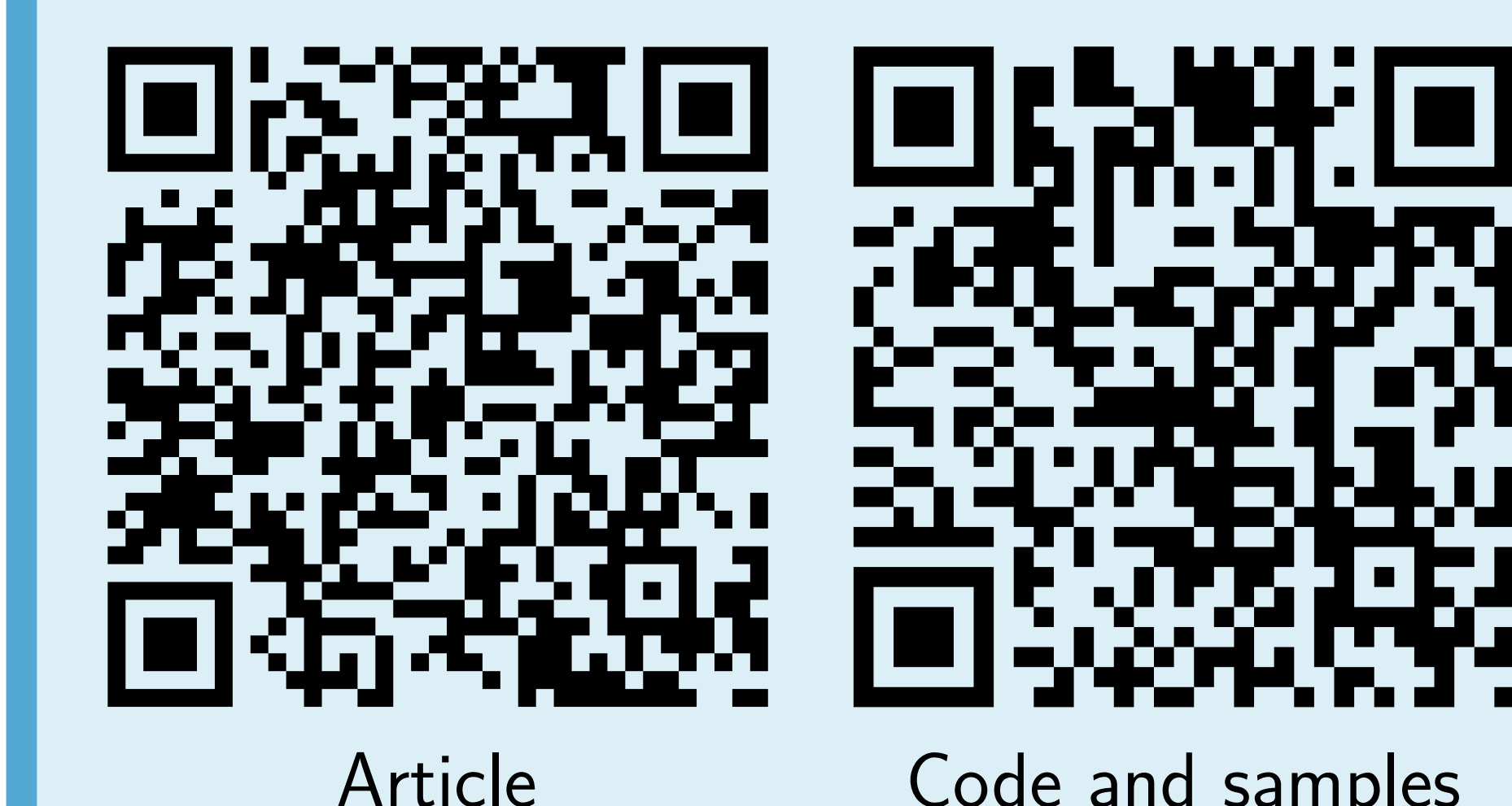
## GANs

- training:  $dg_{\theta_t}(z) = \eta [A_{\theta_t}(z)] (\nabla h_{\rho_t}) dt$
- generation:  $z \sim p_z \rightarrow g_{\theta} \rightarrow x_t$
- discrimination:  $x_t \rightarrow f_{\rho_t} \rightarrow \text{real/fake}$
- Gradient descent-ascent on the min-max objective yields the generator loss:  $\mathcal{L}_{\text{GAN}}(g_{\theta}) = \mathbb{E}_{z \sim p_z} [(c \circ f_{\rho})(g_{\theta}(z))]$ .

## Discr. Flows in Practice

- Discriminator loss:  $\mathcal{L}_d(f; \rho, p_{\text{data}}) = \mathbb{E}_{\rho} [a \circ f] - \mathbb{E}_{p_{\text{data}}} [b \circ f] + \mathcal{R}(f; \rho, p_{\text{data}})$ .
- Naive training: successive  $f_{\rho_t}$  trainings and  $\rho_t$  updates.
- For efficiency purposes, we simultaneously learn all time-parameterized discriminators:  $f_{\rho_t} = f_{\phi}(\cdot, t)$ .
- Training step:
  - sample  $t \sim \mathcal{U}([0, 1])$ ,  $x_0 \sim \pi$ ;
  - compute  $x_t = -\eta \int_0^t \nabla (c \circ f_{\phi}(\cdot, s))(x_s) ds$ ;
  - train  $f_{\phi}(\cdot, t)$  to discriminate between  $x_t$  and  $p_{\text{data}}$ .
- Generalization of some gradient flows.

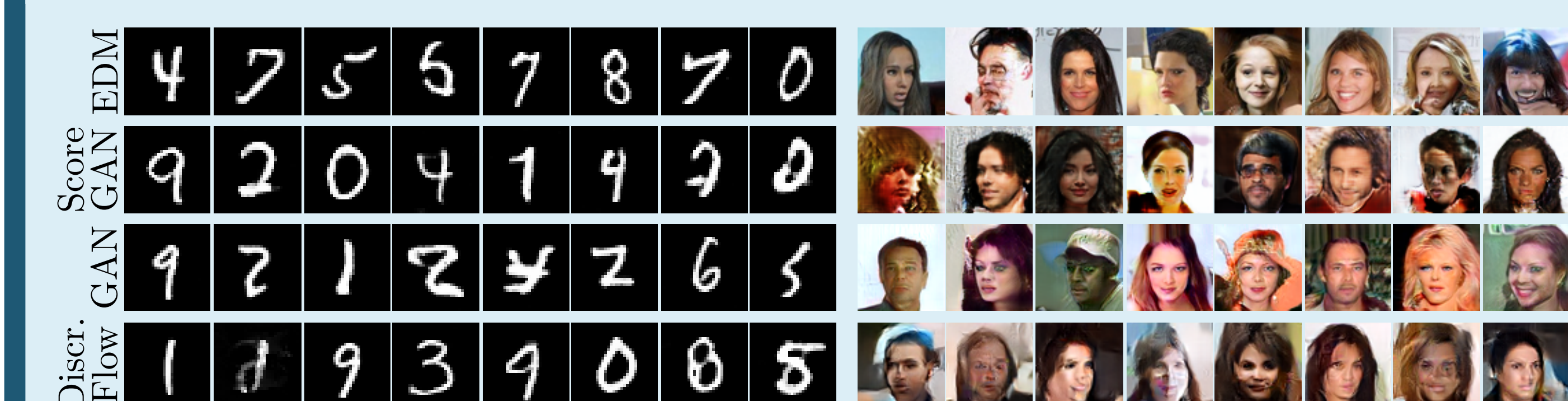
## Links



- Code for all models and baselines.
- Animated samples.

## Experimental Results

Dataset	PMs (no generator)	Int-PMs (generator)
EDM	4	7
Discr. Flow	9	20
GAN	9	21
Score GAN	1	9
EDM	5	7
Discr. Flow	10	41
GAN	3	19
Score GAN	15	35



- Hybrid models are viable, support theory.
- EDM: diffusion (Karras et al., 2022).

## Properties

- PMs vs Int-PMs**: Int-PMs are prone to mode collapse but are faster than PMs at inference and have better latent space properties.
- Discr. flows** learn a path to the data distribution, unlike **diffusion**.

## Perspectives

- Our work paves the way for new hybrid models.
- Model improvements: Score GANs for score distillation, Discr. Flows for generation efficiency.
- Framework improvements: convergence, second-order and discrete-time optimization, more accurate GAN modeling.