Unifying GANs and Score-Based Diffusion as Generative Particle Models

Jean-Yves Franceschi,[1] Mike Gartrell,[1] Ludovic Dos Santos,[1,*] Thibaut Issenhuth,[1,2,*] Emmanuel de Bézenac,[3,*] Mickaël Chen,[4,*] Alain Rakotomamonjy[1,*]
[1]Criteo AI Lab, Paris, France [2]LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallée, France [3]SAM, D-MATH, ETH Zürich, Zürich-8092, Switzerland [4]Valeo.ai, Paris, France

Our Contributions

- We unify gradient flows, score-based diffusion models, and GANs by representing generated data as moving particles.
- A model is defined by:
  - a gradient vector field that the particles follow;
  - the possibility of incorporating a generator smoothing 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).

GANs vs Diffusion

Traditional opposition in the literature.

GANs → Generator trained by discriminating true vs fake data.

- Generator (manifold learning / hypothesis).
- Close to SOTA performance.
- Harder to optimize.
- Fast inference.

Diffusion → Learns to progressively reverse a data degradation process.

- No generator (operates on the data space).
- SOTA performance.
- Easier to optimize.
- Slow inference.

Links

Article

Code and samples

Particle-Based Framework

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

Wasserstein Gradient

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

Log Ratio Gradient

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

Discriminator Gradient

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

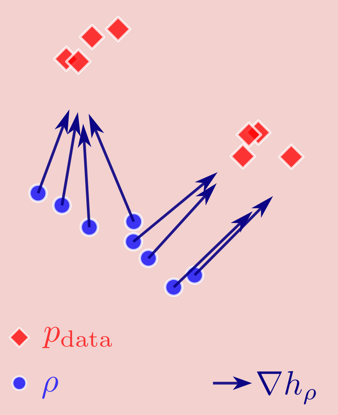
where f_{ρ_t} discriminates ρ_t from p_{data}

Particle Models (No Generator)

- At generation / inference time t:

$$x_0 \sim \pi = \rho_0, \quad \mathrm{d}x_t = \nabla h_{\rho_t}(x_t)\,\mathrm{d}t.$$

- Independently moving particles.
- Each x_t individually follows a **gradient ascent** path on $h_{\rho_t}(x_t)$.
- h_ρ is usually a predefined functional approximated with neural networks.

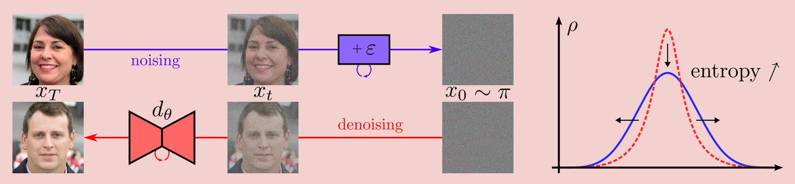


Wasserstein Gradient Flows

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

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

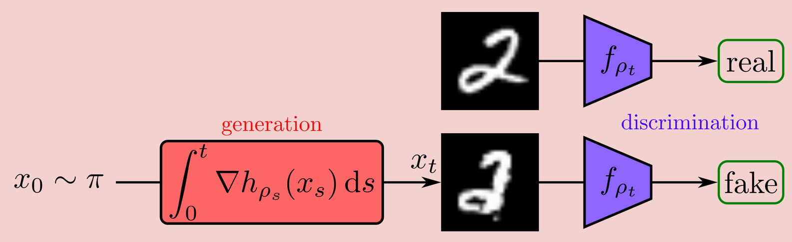
Score-Based Diffusion



- Using Jordan et al. (1998) in Song et al. (2019) and Karras et al. (2022):

$$\mathrm{d}x_t = \alpha_t \nabla \log \left[p_{\text{data}} \star k_{\text{RBF}}^{\sigma(t)}\right](x_t)\,\mathrm{d}t + \sqrt{2\beta_t}\,\mathrm{d}W_t \quad \Leftrightarrow \quad -\beta_t \nabla \log \rho_t\,\mathrm{d}t.$$

Discriminator Flows



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

Interacting Particle Models (Generator)

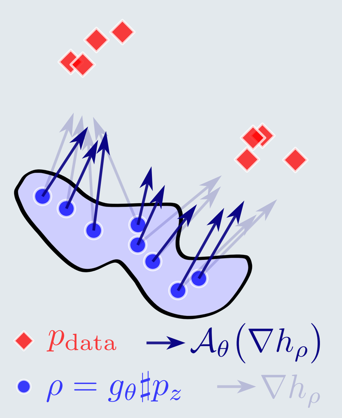
- Training with **the same loss**:

$$\mathcal{L}_{\text{gen}}(\theta) = -\mathbb{E}_{z\sim p_z}\left[h_{\rho_t}\left(g_\theta(z)\right)\right].$$

- At training time t:

$$\mathrm{d}g_{\theta_t}(z) = \eta\left[\mathcal{A}_{\theta_t}(z)\right]\left(\nabla h_{\rho_t}\right)\mathrm{d}t.$$

- Generalization of PMs where particles interact with each other.



Stein Gradient Flows

- Stein gradient flows (Liu, 2017) are kernelized Wasserstein gradient flows:

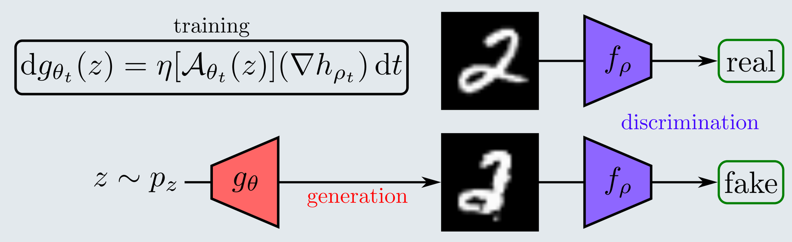
$$\mathrm{d}x_t = \mathbb{E}_{x_t'\sim\rho_t}\left[k\left(x_t, x_t'\right)\nabla h_{\rho_t}\left(x_t'\right)\right]\mathrm{d}t.$$

- Int-PMs under mild hypotheses (generalization of Durr et al. (2022)).
- Hint towards the same h_ρ being used in a PM and an Int-PM.

Score GANs

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

GANs



- Gradient descent-ascent on the min-max objective yields the generator loss:

$$\mathcal{L}_{\text{GAN}}(g_\theta) = \mathbb{E}_{z\sim p_z}\left[\left(c \circ f_\rho\right)\left(g_\theta(z)\right)\right].$$

Smoothing Operator

- $\mathcal{A}_{\theta_t}(z)$ is a linear operator on vector fields (kernel integral operator):

$$\left[\mathcal{A}_{\theta_t}(z)\right](V) \triangleq \mathbb{E}_{z'\sim p_z}\left[k_{g_{\theta_t}}(z, z')V\left(g_{\theta_t}(z')\right)\right],$$

$$k_{g_{\theta_t}}(z, z') \triangleq \nabla_{\theta_t} g_{\theta_t}(z')^\top \nabla_{\theta_t} g_{\theta_t}(z).$$

- $k_{g_{\theta_t}}$ is the generator's 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: $\left[\mathcal{A}_{\theta_t}(z)\right](V) = V\left(g_\theta(z)\right)$.
  - $\mathrm{d}g_{\theta_t}(z) = \nabla h_{\rho_t}\left(g_{\theta_t}(z)\right)\mathrm{d}t$: we retrieve PMs.
- General case: $\mathcal{A}_{\theta_t}$ represents the parameterization of ρ as a manifold.
  - $\mathcal{A}_{\theta_t}$ smooths the original vector field ∇h_{ρ_t} by convolving it with k.
  - Particles interact with each other through generator parameterization.

From PMs to Int-PMs

- We assign to each generated particle $x = g_\theta(z)$ the same loss as in PMs:

$$\mathcal{L}_{\text{gen}}(\theta) = -\mathbb{E}_{z\sim p_z}\left[h_{\rho_t}\left(g_\theta(z)\right)\right].$$

- We do not take into account the dependency of ρ_t w.r.t. θ_t, to mimic PMs: $\rho = \mathrm{StopGradient}(g_\theta \sharp p_z)$.
- Continuous-time gradient descent:

$$\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = -\eta \nabla_{\theta_t} \mathcal{L}_{\text{gen}}(\theta_t) = \eta \nabla_{\theta_t} \mathbb{E}_{z\sim p_z}\left[h_{\rho_t}\left(g_{\theta_t}(z)\right)\right]$$

$$= \eta \mathbb{E}_{z\sim p_z}\left[\nabla_{\theta_t} g_{\theta_t}(z) \nabla h_{\rho_t}\left(g_{\theta_t}(z)\right)\right].$$

- Evolution of particles:

$$\frac{\mathrm{d}g_{\theta_t}(z)}{\mathrm{d}t} = \nabla_{\theta_t} g_{\theta_t}(z)^\top \frac{\mathrm{d}\theta_t}{\mathrm{d}t} = \eta \mathbb{E}_{z'\sim p_z}\left[\nabla_{\theta_t} g_{\theta_t}(z)^\top \nabla_{\theta_t} g_{\theta_t}(z') \nabla h_{\rho_t}\left(g_{\theta_t}(z')\right)\right].$$

Other Models & Flows

- Int-PMs and Stein (generalization of Durr et al. (2022)): $k\left(g_{\theta_t}(z), g_{\theta_t}(z')\right) = k_{g_{\theta_t}}(z, z')$ 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).

Score GANs in Practice

- Two practical issues:
  - sliced score matching to train s_ϕ^ρ;
  - scheduling σs w.r.t. training time t.
- We randomly sample σ 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_ϕ^ρ with denoising score matching;
  - gradient descent step:

$$\theta \leftarrow \theta + \eta \mathop{\mathbb{E}}_{\sigma\sim p_\sigma, \varepsilon\sim\mathcal{N}(0,\sigma I_D), z\sim p_z}\left[\nabla_\theta g_\theta(z) \widetilde{\nabla} h_\rho\left(g_\theta(z) + \varepsilon, \sigma\right)\right].$$

Discriminator Flows in Practice

- Discriminator loss:

$$\mathcal{L}_{\mathrm{d}}(f; \rho, p_{\text{data}}) = \mathbb{E}_\rho[a \circ f] - \mathbb{E}_{p_{\text{data}}}[b \circ f] + \mathcal{R}(f; \rho, p_{\text{data}}).$$

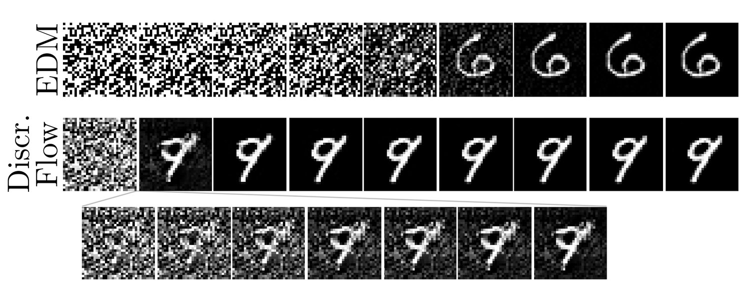
- Naive training: successive f_{ρ_t} trainings and ρ_t updates.
- For efficiency purposes, we simultaneously learn all time-parameterized discriminators: $f_{\rho_t} = f_\phi(\cdot, t)$.
- Training step:
  - sample $t \sim \mathcal{U}([0, 1])$, $x_0 \sim \pi$;
  - compute $x_t = -\eta \int_0^t \nabla\left(c \circ f_\phi(\cdot, s)\right)(x_s)\,\mathrm{d}s$;
  - train $f_\phi(\cdot, t)$ to discriminate between x_t and p_{data}.
- Generalization of some gradient flows.

Experimental Results

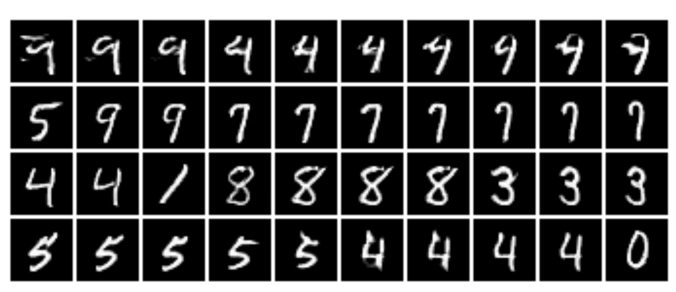
Dataset	PMs (no generator)		Int-PMs (generator)	
	EDM	Discr. Flow	GAN	Score GAN
MNIST	3	4	3	15
CelebA	10	41	19	35

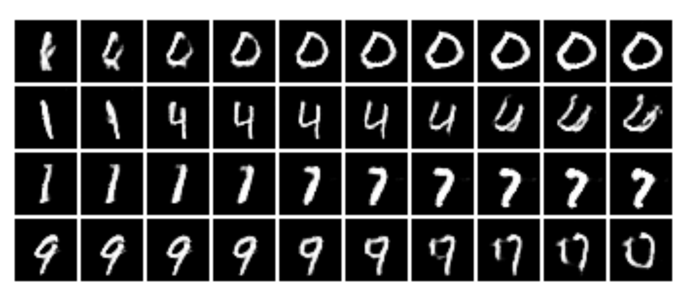
- Hybrid models are viable, and support the theory.
EDM: diffusion (Karras et al., 2022).

Properties



- **Discriminator flows** learn a path to the data distribution, unlike **diffusion**.





- **PMs vs Int-PMs**: Int-PMs are prone to mode collapse but are faster than PMs at inference and have better latent space properties.

Perspectives

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

NeurIPS@Paris 2023, Paris, France