Proximal divergences and generative modeling

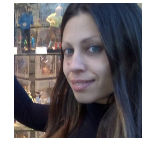
Luc Rey-Bellet

ICERM: Robust Optimization and Simulation of Complex Stochastic Systems

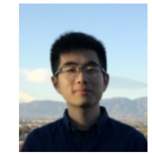
2024-09-13

Team supported by NSF and AFOSR

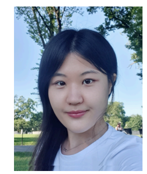
- Paul Dupuis (Brown University)
- Markos Katsoulakis (UMass Amherst)


Panagiota

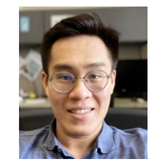

Jeremiah

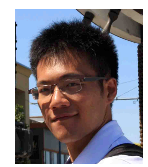

Ziyu

- Panagiota Birmpa (UMass ⟶ Heriot Watt)
- Jeremiah Birrell (UMass ⟶ Texas State San Marco)
- Ziyu Chen (UMass Amherst)
- Hyemin Gu (UMass Amherst)
- Yannis Pantazis (FORTH, Crete)
- Benjamin Zhang (UMass Amherst ⟶ Brown)
- Wei Zhu (UMass Amherst ⟶ Georgia Tech)


Hyemin


Yannis


Benjamin


Wei

Some papers relevant to the talk

- J. Birrell, P. Dupuis, M. A. Katsoulakis, L. Rey-Bellet, J. Wang, *A Variational Formula for Rényi Divergences*, SIAM Data Science, (2021).
- P. Dupuis and Y. Mao, *Formulation and properties of a divergence used to compare probability measures without absolute continuity*, ESAIM: COCV, (2022).
- J. Birrell, M.A. Katsoulakis, L. Rey-Bellet, W. Zhu, *Structure-preserving GANs*. ICML 2022
- J. Birrell, P. Dupuis, M. A. Katsoulakis, Y. Pantazis, L. Rey-Bellet, *(f, Γ) -Divergences: Interpolating between f-Divergences and Integral Probability Metrics*, JMLR & NeurIPS, (2022)
- J. Birrell, P. Dupuis, M. A. Katsoulakis, Y. Pantazis, L. Rey-Bellet, *Function-space regularized Rényi divergences*, ICLR 2023
- Z. Chen, M. A. Katsoulakis, L. Rey-Bellet, W. Zhu, *Sample Complexity of Probability Divergences under Group Symmetry*, ICML 2023
- H. Gu, P. Birmpa, Y. Pantazis, M. A. Katsoulakis, and L. Rey-Bellet, *Lipschitz-regularized gradient flows and generative particles*, SIAM Data Science (2024), to appear.
- Z. Chen, M. A. Katsoulakis, L. Rey-Bellet, W. Zhu, *Statistical Guarantees of Group-Invariant GANs*, ArXiv, (2023).
- H. Gu, M. A. Katsoulakis, L. Rey-Bellet, B. Zhang, *Combining Wasserstein-1 and Wasserstein-2 proximals: robust manifold learning via well-posed generative flows*, ArXiv (2024)
- Z. Chen, H. Gu, M. A. Katsoulakis, L. Rey-Bellet, W. Zhu, *Learning heavy-tailed distributions with Wasserstein-proximal-regularized α-divergences* , ArXiv (2024)

Goals of generative modeling

- Given data $(X_i)_{i=1}^N$ with $X_i \sim \pi$ with π unknown typically on $\mathbb{R}^d$ with $d \gg 1$.
- Generative modeling = learn a representation of the random variable X
  - Pick a source ρ (easy to simulate)
  - Learn a generative map Φ such that $\Phi_{\#}\rho = \pi$.
    - Alternatively learn a transport plan (i.e. a Markov kernel).
  - Or learn a generative flow via a vector field $v_t(x)$ such that

$$dx_t = v_t(x_t) + \sigma_t dW_t \quad \text{such that} \quad x_0 \sim \rho \quad \text{and} \quad x_T \sim \pi$$

    - $\sigma = 0$ learn an ODE (normalizing flows, neural ODEs, etc…)
    - $\sigma > 0$ learn an SDE (diffusion models, score generative model, Schrödinger bridges, etc…)

Generative modeling = information theoretic task

In oder to learn how to transport ρ to π (and so do the learning) we need to choose how to measure the “distance” between ρ and π.

- KL divergence (or more generally f-divergences $x\ln(x) \to f(x)$ convex, see the papers, typical is $\frac{x^\alpha - 1}{\alpha(\alpha-1)}$)

$$D_{\mathrm{KL}}(\rho\|\pi) = E_\pi\left[\frac{d\rho}{d\pi}\ln\frac{d\rho}{d\pi}\right] = \sup_{\phi\in C_b(X)}\left\{E_\rho[\phi] - \log E_\pi[e^\phi]\right\}$$

  - Good: Maximum likelihood and all that plus an excellent convex dual formula (Gibbs variational formula)
  - Bad: Restricted to $\rho \ll \pi$ which is not adequate in ML: π is often supported on low-dimensional structure and π is known via its empirical distribution $\pi_N = \frac{1}{N}\sum_i \delta_{X_i}$!

- Integral probability metrics (IPM): pick a set $\Gamma \in C_b(\mathcal{X})$ which is convex and closed (weak* topology) such that $f \in \Gamma \implies -f \in \Gamma$ and Γ separate points in $\mathcal{P}(X)$.

$$W^{\Gamma}(\rho, \pi) = \sup_{\phi \in \Gamma}\{E_\rho[\phi] - E_\pi[\phi]\}$$

  - Bad: The optimization problem is "too linear", not convex enough.
  - Good: Γ is often a very good set to optimize over!
    - Γ = L-Lipschitz functions
    - Γ = Unit ball in RKHS $\longrightarrow$ Kernel methods and MMD distance (Hilbert space embedding of $\mathcal{P}(X)$)
    - Γ = Sets of Relu Neural Networks with spectral normalization ($\longrightarrow$ bound on the Lipschitz constant!): Neural IPMs

- Optimal transport: Wasserstein distances: given a weight. e.g., $c(x,y) = \|x-y\|^p$

$$W_p^p(\rho,\pi) = \inf_{\gamma \text{ coupling}} \int_{X\times X} c(x,y)d\gamma(x,y) = \sup_{\phi(x)+\psi(y)\le c(x,y)} E_\rho[\phi] + E_\pi[\psi]$$

  - Bad: Costly to compute the optimal transport map/plan and optimization is “too linear”. Optimal is not optimal!. Sinkhorn (Schrödinger bridges) is a popular tool.
  - Good: There is an implicit regularization. In the dual formula the supremum can restricted to $\phi = \psi^c$ where

$$\psi^c(x) = \inf_y \{c(x,y) + \phi(x)\} \quad c-\textbf{transform}$$

For example for $p=1, W_1(\rho,\pi)$ is the IPM with 1-Lipschitz function.

  - Good: Benamou-Bremier representation (for $p>1$) $\to$ Flows!

Moreau-Yosida regularization a.k.a inf-convolution

How to regularize a (convex) function?

$$f \Box g(x) = \inf_y \{f(y) + g(x-y)\} \quad \text{infimum convolution}$$

Examples:

- $f_L(x) = \inf_y\{f(y) + L\|x-y\|\}$ is L-Lipschitz and $\lim_{L\to\infty} f_L(x) = f(x)$.
- $f_\lambda(x) = \inf_y\{f(y) + \lambda\|x-y\|^2\}$ makes f finite, smooth, and preserves convexity $\longrightarrow$ proximal optimization algorithms
- $f^c(x) = \inf_y\{f(y) + c(x,y)\}$ is the c-transform, key regularizing tool in optimal transport e.g in Kantorovich-Rubinstein duality (provides compactness!)

Proximal IPM divergences

Use an IPM to regularize KL-divergence (Dupuis, Mao) or general f divergences (Birell et al.)

$$D^{\Gamma}_{KL}(\rho\|\pi) = \inf_{\mu \in \mathcal{P}(X)} \left\{ W^{\Gamma}(\rho,\mu) + D_{KL}(\mu\|\pi) \right\}$$

Elementary properties:

1. $D^{\Gamma}_{KL}(\rho\|\pi) \le \min\{W^{\Gamma}(\rho,\pi), D_{KL}(\rho\|\pi)\}$ so no absolute continuity needed!
2. Using the compactness and strict convexity of the level sets of $\rho \mapsto D_{KL}(\rho,\pi)$ there is a unique optimizer μ^*

$$D^{\Gamma}_{KL}(\rho\|\pi) = W^{\Gamma}(\rho,\mu^*) + D_{KL}(\mu^*\|\pi)$$

This define a proximal operator $\mu^* = \mathbf{prox}_{D_{KL}}(\rho)$

3. Interpolation: If we scale Γ with $\Gamma_L = L\Gamma$ we have

$$\lim_{L\to\infty} D_{KL}^{\Gamma_L}(\rho\|\pi) = D_{KL}(\rho\|\pi)$$

$$\lim_{L\to 0} \frac{1}{L} D_{KL}^{\Gamma_L}(\rho\|\pi) = W^{\Gamma}(\rho,\pi)$$

How to pick L?

- Sometimes in ML, the proximal $\mu^* = \mathrm{prox}_{D_{KL}}(\rho)$ will serve as the model for π so we should adjust L accordingly that is L small enough.
- In other cases we choose L to stabilize the learning algorithm (often $L = 1$.)
- More theory needed: convergence of the proximal

Variational principle for proximal IPM divergences

Theorem 1 (Gibbs variational principle)

$$D_{KL}^{\Gamma}(\rho\|\pi) = \inf_{\mu\in\mathcal{P}(\mathcal{X})}\left\{W^{\Gamma}(\rho,\mu) + D_{KL}(\mu\|\pi)\right\}$$
$$= \sup_{\phi\in\Gamma}\left\{E_{\rho}[\phi] - \log E_{\pi}[e^{\phi}]\right\}$$

Proof: With $I_{\Gamma}(\phi) = \infty 1_{\Gamma^c}(\phi)$ to impose the constraint and the fact that for Legendre transform $(f+g)^* = f^* \square g^*$ we find (using the duality pair $(C_b(X), \mathcal{M}(X))$).

$$\sup_{\phi\in\Gamma}\left\{E_{\rho}[\phi] - \log E_{\pi}[e^{\phi}]\right\} = \sup_{\phi\in C_b(X)}\left\{E_{\rho}[\phi] - \log E_{\pi}[e^{\phi}] + I_{\Gamma}(\phi)\right\}$$
$$= (\log E_{\pi}[e^{\phi}] + I_{\Gamma}(\phi))^* = \log E_{\pi}[e^{\phi}]^* \square I_{\Gamma}(\phi)^*$$
$$= (D_{KL} \square W^{\Gamma})(\rho)$$

One also obtains the following characterization of the proximal μ^* (for Γ=Lipschitz)

Theorem 2 For Γ L-Lipschitz, if

$$\phi^* = \arg\max_{\phi \in \Gamma} \left\{ E_\rho[\phi] - \log E_\pi[e^\phi] \right\}$$

and

$$\mu^* = \arg \min_{\mu \in \mathcal{P}(X)} \{ W^\Gamma(\rho, \mu) + D_{KL}(\mu \| \pi) \} \quad \text{proximal}$$

then we have

$$\frac{d\mu^*}{d\pi} = e^{\phi^*}$$

This captures the balance between transport (done by IPM) and mass redistribution (done by KL).

Proximal OT divergences

$$D_{KL}^{p,\lambda}(\rho\|\pi) = \inf_{\mu \in \mathcal{P}(X)} \left\{ \lambda W_p^p(\rho, \mu) + D_{KL}(\mu\|\pi) \right\}$$

For $p = 1$ this an IPM but for $p > 1$ this regularizes with the p^{th} power of the Wasserstein distance. Here $p = 2$ for illustration.

By the Benamou-Bremier representation of optimal transport

$$D_{KL}^{2,\lambda}(\rho\|\pi) = \inf_{\rho,v} \left\{ D_{KL}(\rho\|\pi) + \lambda \int_0^1 E_{\rho_t}\left[\frac{1}{2}\|v_t\|^2\right] \right\}$$
$$\text{subject to} \quad \partial_t \rho_t + \nabla \cdot (v_t \rho_t) = 0, \quad \rho_0 = \rho$$

This is good way to build generative flows (see later in the talk)

Gibbs variational principle for Wasserstein proximal

There is also a dual formula (not used further today)

Theorem 3 For general weights $c(x, y)$ (bounded below and lower semicontinous) and X a Polish space we have the duality formula

$$D_{KL}^{C}(\rho\|\pi) = \inf_{\mu\in\mathcal{P}(X)} \{W^{c}(\rho,\mu) + D_{KL}(\mu\|\pi)\}$$
$$= \sup_{\phi(x)+\psi(y)\leq c(x,y)} \{E_{\rho}[\phi] - \log E_{\pi}[e^{-\psi}]\}$$

- This divergences have nice properties, similar to proximal IPM (for another day).
- See Jeremiah Birrell for similar results and applications to DRO!

Generative adversarial networks (GANS)

Birell et. al (JMLR 2022)

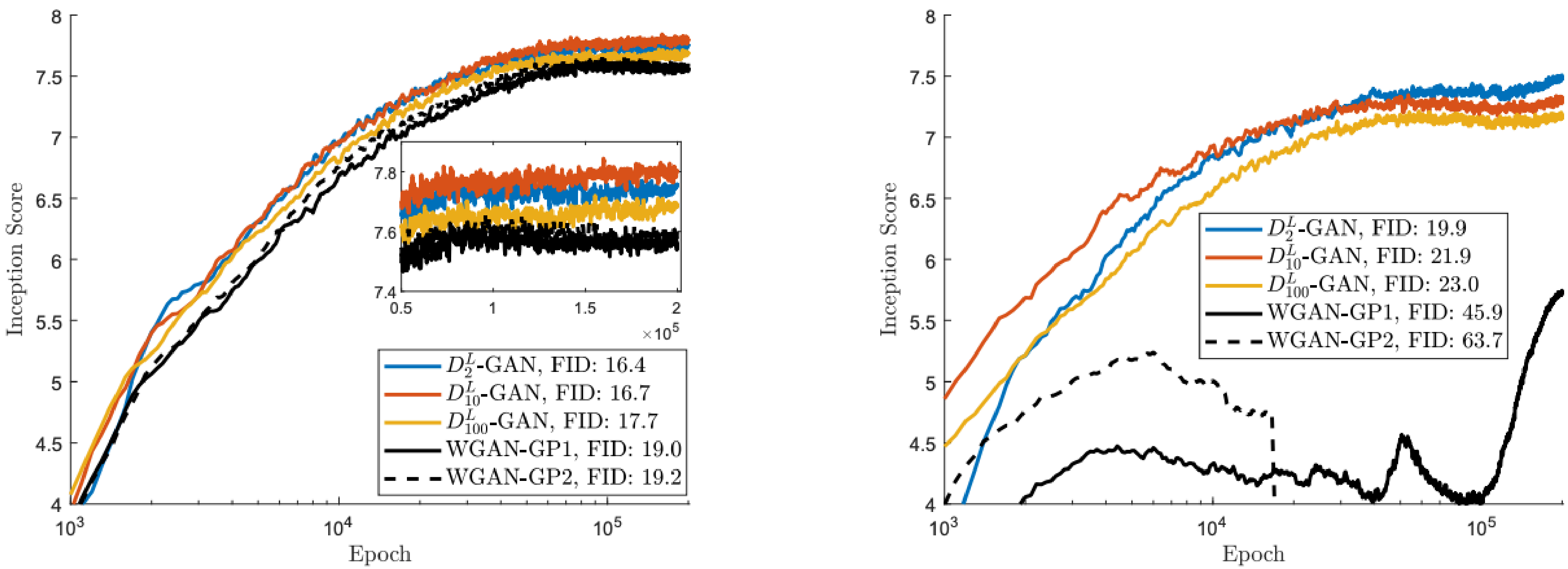
- Choose a reference space (Ω_{ref}, ρ) (usually Gaussian, low-dimensional) and an objective functional (usually a probability divergence).
- Optimization problem $(KL - \Gamma)$-GAN

$$\inf_g D(g_{\#}\rho \| \pi) = \inf_g \sup_{\phi \in \Gamma} \left\{ E_\pi[\phi] - \log E_{g_{\#}\rho}[e^\phi] \right\}$$

  - Optimization over maps $g : \Omega_{ref} \to X$ (parametrized by suitable neural networks) provides the generative model $\mu = g_{\#}\rho$ which approximates π.
  - Solve via min-max algorithms
  - Replacing π and ρ by corresponding their empirical measure (and mini-batches).

Findings:

- Provide a natural and theoretically grounded way to stabilize the training of f-GAN.: Proximal IPM divergences incorporate the Lipschitz regularization of neural networks (via spectral normalization or soft constraints) into the divergence.
- Empirically, that KL-Lischitz GAN outperform Wasserstein GANs ([more robust, less sensitive to choise of hyper parameters and learning rates}{.red}) Intuitively the objective functional is much more (strictly) convex so better convergence of the algorithms is expected. A proof of this would be nice!
- f-GAN (for suitable choices of f) perform very well for heavy tail data (go talk to Ziyu and see his poster)



f-Gan is more stable with respect to learning rates than W-Gan (CIFAR-10 data sets)

First variation of proximal divergences

- Infimum convolution has a smoothing effect:
- The KL-divergence has a well defined first variation

$$\frac{\delta D_{KL}}{\delta \rho}(\rho \| \pi) = \arg\sup_{\phi} \left\{ E_\rho[\phi] - \log E_\pi[e^\phi] \right\} = \phi^* = \log \frac{d\rho}{d\pi}$$

Theorem 4 The Γ-KL proximal divergence has a well defined first variation.
If $\phi^* = \arg\sup_{\phi \in \Gamma} \left\{ E_\rho[\phi] - \log E_\pi[e^\phi] \right\}$ (unique on $\mathrm{supp}(\rho + \pi)$) then

$$\frac{\delta D_{KL}^\Gamma}{\delta \rho}(\rho \| \pi) = \inf_y \{\phi^*(y) + \|x - y\|\} = \overline{\phi}^* \qquad \text{Lipschitz regularization}$$

which is defined for all x.

A similar result holds for Wasserstein proximals.

Wasserstein gradient flow

With the first variation we can consider Wasserstein gradient flow

$$\partial_t \rho_t = \mathrm{div}\left(\rho_t \nabla \frac{\delta D_{KL}^{\Gamma}}{\delta \rho}(\rho_t \| \pi)\right)$$

which we can think as a Lipschitz regularization of the Fokker-Planck equation

We do not need to assume densities which leads to Particle Algorithms which are very well suited for learning tasks from data.

Gradient Particle algorithm Given data $X_i \sim \pi$ and source samples $Y_j \sim \rho$ Euler method gives

$$Y_{j,n+1} = Y_{j,n} - \Delta t \nabla \phi_n^*(Y_{j,n})$$

$$\phi_n^* = \underset{\phi \in \Gamma_L^{NN}}{\mathrm{argmax}} \left\{ \frac{1}{M} \sum_{i=1}^{M} \phi(Y_{i,n}) - \log \frac{1}{N} \sum_{i} e^{\phi(X_i)} \right\}$$

- Since ϕ^* is Lipschitz we have finite speed propagation (CFL-type condition) $\longrightarrow$ stability of the numerical schemes.
- The gradient structure implies that

$$\frac{d}{dt} D_{KL}^{\Gamma_L}(\rho_t \| \pi) = -I_f^{\Gamma_L}(\rho_t \| \pi) \leq 0$$

where we define the Lipschitz-regularized Fisher Information as

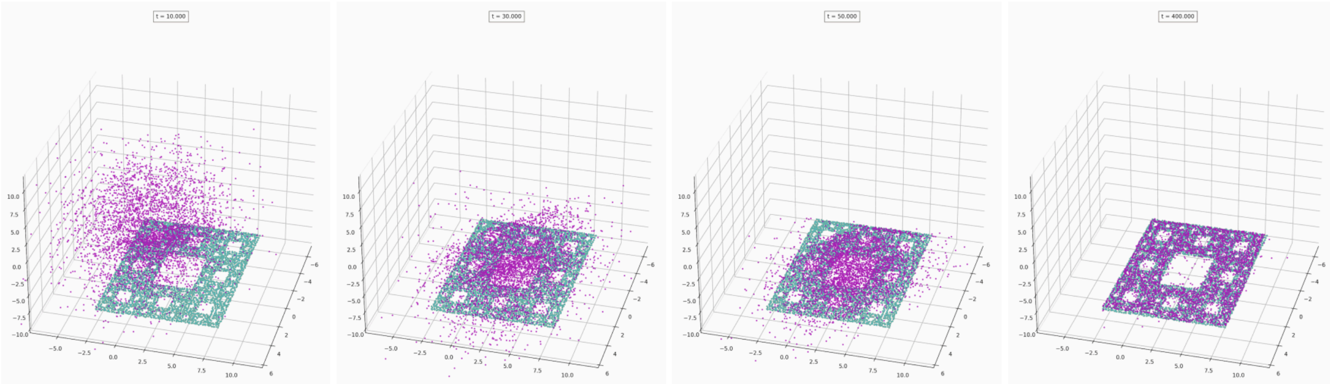
$$I_{KL}^{\Gamma_L}(\rho_t \| \pi) = E_{\rho_t}\left[|\nabla \phi^*|^2 \right] .$$

For particles this is just the total kinetic energy of the particles

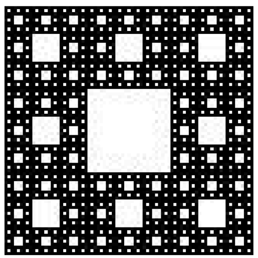
$$I_{KL}^{\Gamma_L}(\hat{\rho}_n^M \| \hat{\pi}^N) = \frac{1}{M} \sum_{i=1}^{M} |\nabla \phi_n^{L,*}(Y_n^{(i)})|^2 ,$$

D_{KL}^{Γ} and the Fisher information $I_{KL}^{\Gamma_L}$ can be monitored to ensure convergence.

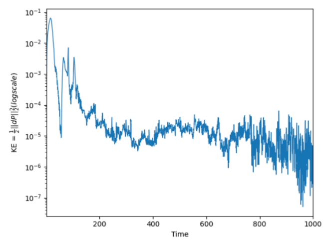
Sierpinksi carpet



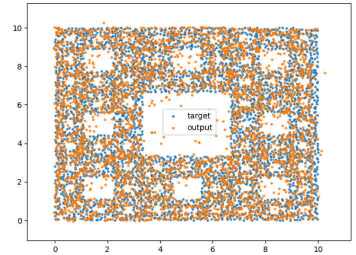
GPA



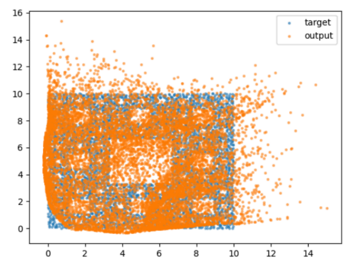
(a) Target distribution



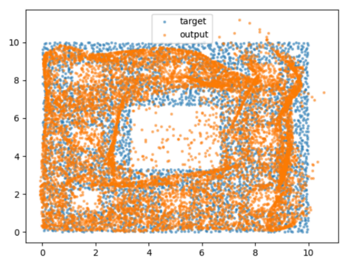
(b) Kinetic energy of particles eq. (33) for $(f_{\mathrm{KL}}, \Gamma_1)$-GPA



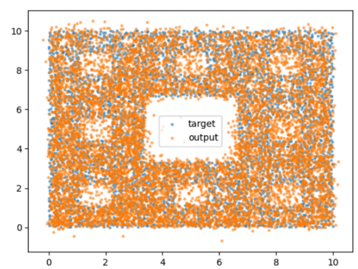
(c) Output of $(f_{\mathrm{KL}}, \Gamma_1)$-GPA



(d) Output of WGAN [4]



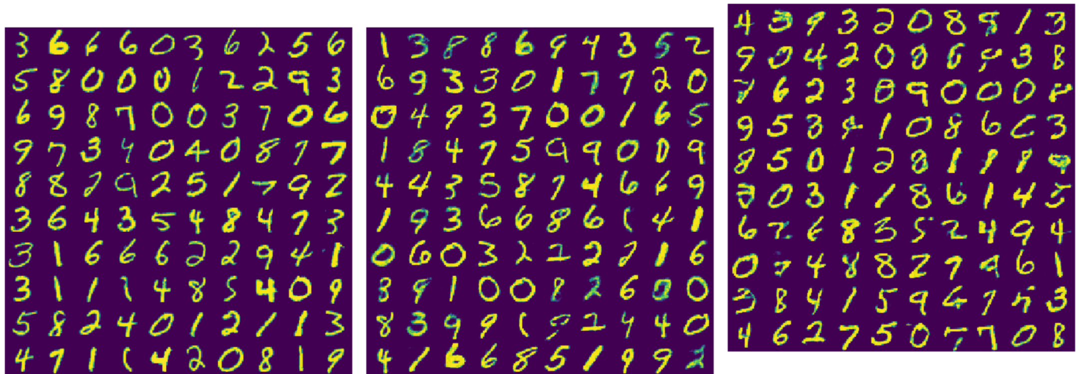
(e) Output of $(f_{\mathrm{KL}}, \Gamma_1)$-GAN [5]



(f) Output of SGM [58]

The other guys

MNIST with scarce data and generalization



(a) Fixed target samples with sample size $N = 200$

(b) $M = 600$ transported particles from $(f_{\mathrm{KL}}, \Gamma_5)$-GPA

(c) 600 generated particles that are simultaneously transported from $(f_{\mathrm{KL}}, \Gamma_5)$-GPA

Figure 5: **(MNIST) GPA for image generation given scarce target data.** **(a)** A subset of the $N = 200$ target

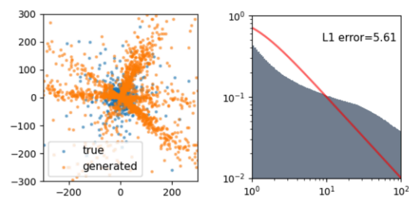
(a) WGAN [4] trained with 200 original data for 3000 training epochs

(b) WGAN [4] trained with original 1400 data for 500 training epochs

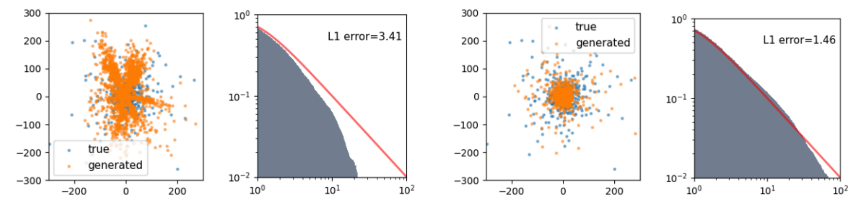
(c) WGAN [4] trained with 200 original data and 1200 GPA-augmented data for 500 training epochs

Heavy-tailed Distributions (Ziyu's poster)

- Learn heavy-tailed distributions using generative models
- Theory in Ziyu's poster!
- GPA and Γ-GANS perfom best compared to other generative algorithms

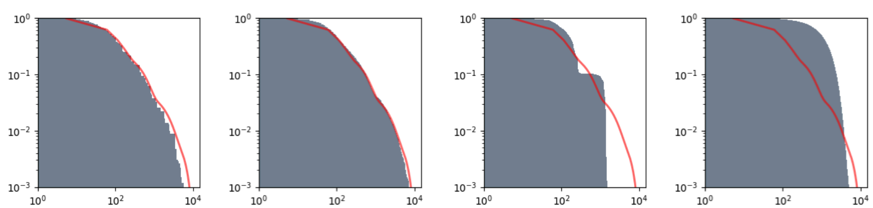


(a) CNF



(b) OT flow (c) VE-SGM

Figure 4: Learning a 2D isotropic Student-t with degree of freedom $\nu = 1$ (tail index $\beta = 3.0$) using generative models based on W_2-α-divergences with $\alpha = 1$. Models with W_2-proximal regularizations, (b) and (c), learn the heavy-tailed distribution significantly better than that without, (a). See Section 5.1 for detailed explanations of the models.



(a) W_1-α GPA (left), W_1-α GAN (right) (b) W_2 flow (left), VE SGM (right)

Figure 5: Sample generation of inter-arrival time between keystrokes. Generative models with W_1-proximal regularization, panel (a), outperform those with W_2-proximal regularization, panel (b), in capturing the tails. This observation suggests that W_1-proximal algorithms can potentially handle heavier tails more effectively than W_2-proximal methods.

Normalizing flows

Continuous normalizing flows (many different variants) train ODE's

$$\frac{dx_t}{dt} = v_t(x_t) \quad \text{with } x_0 \sim \pi \text{ and } x_1 \sim \rho$$

by minimizing $D_{KL}(\pi | g_{\#}\rho)$ where g=time-1 map. Use the change of variables for densities to evaluate KL.

- One need to invert the flow to generate π from ρ (backward-forward flows).
- The training is unstable and depends on the time discretization.
- Autoencoder and specialized archtecture are needed.
- For target π which are singular the use of densities is a bit suspicious.

$W_1 + W_2$ proximal (Hyemin's poster)

Main ideas:

- Use Benamou-Bremier and W_2^2 proximal to stabilize the learning of the flow
- Replace {D_{KL} by D_{KL}^{Γ} to handle singular π.]{.red}

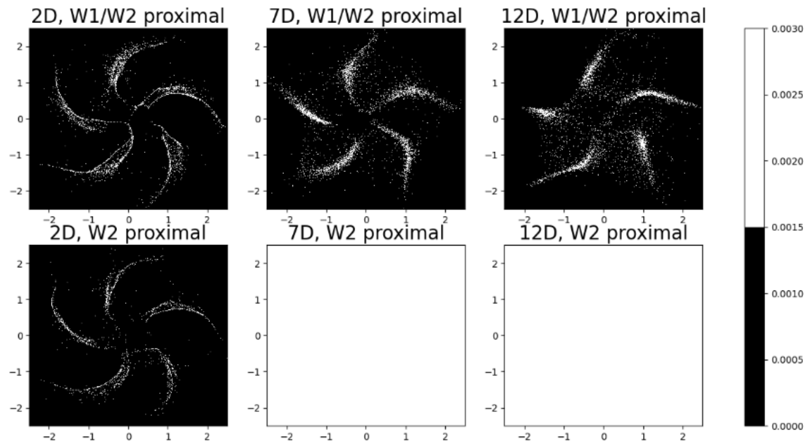
Putting all together we find the functional

$$\inf_{v,\mu} \left\{ \sup_{\phi \in \Gamma_L} \left\{ \mathbb{E}_\mu[\phi] - \log E_\pi[e^\phi] \right\} + \lambda \int_0^1 \frac{1}{2} E_{\rho_t}[|v_t|^2] \right\}$$

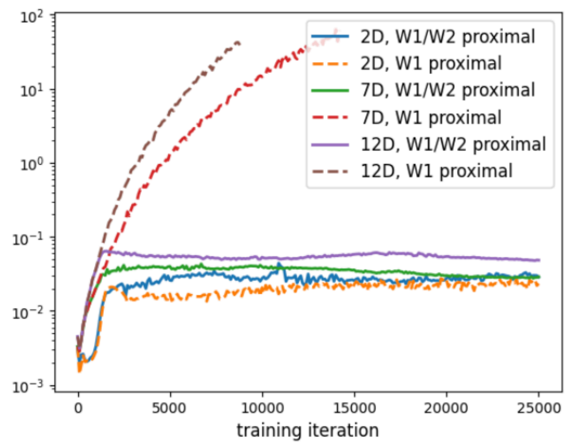
$$\text{subject to} \quad \frac{dx_t}{dt} = v_t(x_t),\ x_0 \sim \rho, x_1 \sim \mu$$

- Adversarial training like in GANs so no need to invert the flow.
- Capture high dimensional strucutre without auto-encoder!

Example: capturing low-d structure



(a) 20K generated samples from $\mathcal{W}_1 \oplus \mathcal{W}_2$-flows (top), $\mathcal{W}_2$-flows (bottom)



(b) Optimality indicator (44)

Dataset	$\mathcal{W}_1 \oplus \mathcal{W}_2$ flow	$\mathcal{W}_2$ flow	Potential Flow GAN [23]	OT flow [17]
Pinwheel 2D	0.00852	**0.00691**	0.01325	0.19793
Pinwheel 7D	**0.01074**	-	16.88652	4.5831e+09
Pinwheel 12D	**0.01662**	-	3.76265	7.9118e+26
Moons 2D	**0.08768**	0.26356	10.11568	2.51535
Moons 7D	**0.02986**	-	221.65057	3.4141e+06
Moons 12D	**0.05259**	-	2229.81445	1.6721e+14

Table 1: Wasserstein-2 distance [8] between the original 2D data manifold and generated 2D data manifold. 5K samples are chosen from the original dataset and the generated dataset. Unlike Potential

Mean-field game analysis

Markos' talk: the optimzation is a mean-field game with optimality conditions in the form of a forward Fokker-Planck equation and a backward Hamilton-Jacobi equation:

$$\partial_t U_t + \frac{1}{2\lambda}|\nabla U_t|^2 = 0 \qquad \text{with} \qquad U_1(x) = \frac{\delta D_f^\Gamma}{\delta \rho}(\mu\|\pi)$$

$$\partial_t \rho_t - \nabla \cdot \left(\rho_t \frac{\nabla U_t}{\lambda}\right) = 0 \qquad \text{with} \qquad \rho_0 = \rho.$$

and with optimal velocity $v_t(x) = -\frac{1}{\lambda}\nabla U_t(x)$.

Theorem 5

- W_1 proximal implies that we have well-defined terminal condition for HBJ + uniqueness of classical solution
- W_2 proximal provides a meaningful PDE + linear optimal trajectories

JKO + Wasserstein gradient flow

Wasserstein gradient flow for $D^{\Gamma}_{KL}(\rho\|\pi)$

$$\partial_t \rho_t = \operatorname{div}\left(\rho_t \nabla \frac{\delta D^{\Gamma}_{KL}}{\delta \rho}(\rho_t\|\pi)\right)$$

= regularized Fokker-Planck

- Explicit Euler = GPA algorithms!

- Implicit Euler = W_1 + W_2 proximal!

Conclusion

We need more good ideas from Paul for many years to come!