Mathematics on Deep Generative Models

Yuling Jiao

School of AI, Wuhan University

Workshop on Foundation and Future of Generative Models: Mathematics, Algorithms, and Applications

CityU, July, 28-30, 2025

Outline

Why deep learning works?

- Alchemy ?

- Data, computational power.
- Learning high-dim functions, distributions with deep nonparametric model.
  - Supervised learning, semi-supervised learning, generative learning, reinforcement learning, representation learning...
    - approximation, generalization, optimization.

Deep feedforward neural network

- FNN
  - $u_\theta(x) = \mathcal{T}_{\mathcal{D}} \circ \sigma_{\mathcal{D}-1} \circ \mathcal{T}_{\mathcal{D}-1} \circ \sigma_{\mathcal{D}-2} \circ ... \mathcal{T}_2 \circ \sigma_1 \circ \mathcal{T}_1(x)$.
  - $\mathcal{T}_i : \mathbb{R}^{w_i} \to \mathbb{R}^{w_{i+1}}, i = 1, ..., \mathcal{D}, \theta_i = (A_i, b_i)$, i.e., $\mathcal{T}_i(x) = A_i x + b_i$
  - σ_i are activate functions.
    - Logistic: $\sigma(z) = \frac{1}{1+e^{-z}}$, Hyperbolic tangent: $\sigma(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$, Rectified linear: $\sigma(z) = \max\{0, z\}$, floor, exponential...
  - $u_\theta(x)$ is neural network with depth $\mathcal{D}$, width $\mathcal{W} = \|w\|_\infty$, size $\mathcal{S} = \sum_{\ell=1}^{\mathcal{D}} w_\ell * w_{\ell+1} + w_{\mathcal{D}}$:
  - Training by stochastic gradient descent Robbins-Monro 51, Bottou-LeCun 05.
    - $\hat{\theta} \in \arg\min_\theta \frac{1}{n} \sum_{i=1}^n \text{loss}(u_\theta(x_i), y_i)$ with tricks: gradient clipping, drop out, bach normalization, spectral normalization...

Working example 1: deep Ritz method (DRM) E-Yu 17

- Consider elliptic PDE with Neumann boundary condition

$$\begin{cases} -\triangle u + wu = f \text{ in } \Omega \\ \dfrac{\partial u}{\partial n} = g \text{ on } \partial\Omega, \end{cases} \tag{1}$$

where $\Omega = (0,1)^d$ is a bounded open subset of $\mathbb{R}^d$, $d > 1$, $f(x) \in L^2(\Omega)$, $w(x) \in L^\infty(\Omega)$ satisfying $w(x) \geq c_1 > 0$ a.e., and $g(s) \in H^{1/2}(\partial\Omega)$.

- u^* solvers (1) iff $u^* \in \arg\min_{u \in H^1(\Omega)} \mathcal{L}(u)$, with

$$\begin{aligned} \mathcal{L}(u) = |\Omega| \mathbb{E}_{X \sim \mathrm{U}(\Omega)} [\|\nabla u(X)\|_2^2/2 + w(X)u^2(X)/2 - u(X)f(X)] \\ - |\partial\Omega| \mathbb{E}_{Y \sim \mathrm{U}(\partial\Omega)} [Tu(Y)g(Y)]. \end{aligned}$$

- Let $\{X_i\}_{i=1}^{N_\Omega}$ i.i.d $\sim \mathrm{U}(\Omega)$, $\{Y_j\}_{j=1}^{N_{\partial\Omega}}$ i.i.d $\sim \mathrm{U}(\partial\Omega)$ and

$$\begin{aligned} &\widehat{u}_\theta \in \arg\min_{u_\theta \in \mathcal{F}_{NN}} \widehat{\mathcal{L}}(u_\theta) \\ &= \frac{|\Omega|}{N_\Omega} \sum_{i=1}^{N_\Omega} [\frac{\|\nabla u_\theta(X_i)\|_2^2}{2} + \frac{w(X_i)u_\theta^2(X_i)}{2} - u_\theta(X_i)f(X_i)] - \frac{\partial\Omega}{N_{\partial\Omega}} \sum_{j=1}^{N_{\partial\Omega}} [u_\theta(Y_j)g(Y_j)]. \end{aligned}$$

- Call a (random) solver $\mathcal{A}$ to solve $\widehat{\mathcal{L}}(u_\theta)$ and output $u_{\theta_{\mathcal{A}}} \in \mathcal{F}_{NN}$.

-

$$\|u_{\theta_{\mathcal{A}}} - u^*\|_{H^1(\Omega)}^2 \leq C[\underbrace{\inf_{\bar{u} \in \mathcal{F}_{NN}} \|\bar{u} - u^*\|_{H^1(\Omega)}^2}_{\mathcal{E}_{\text{app}}} + 2 \underbrace{\sup_{u \in \mathcal{F}_{NN}} |\mathcal{L}(u) - \widehat{\mathcal{L}}(u)|}_{\mathcal{E}_{\text{sta}}} + \underbrace{\widehat{\mathcal{L}}(u_{\theta_{\mathcal{A}}}) - \widehat{\mathcal{L}}(\widehat{u}_\theta)}_{\mathcal{E}_{\text{opt}}}].$$

Working example 2: deep binary classification

- Given observed i.i.d data $\mathbb{D} = \{(X_i, Y_i)\}_{i=1}^n$ with $(X_i, Y_i) \sim \mathbb{P}(x, y)$ where $Y_i = \{1, -1\}$ is the label, $X_i \in \mathbb{R}^d$ is a d-dimensional vector, we want to use ERM for classification.
- Define $u^* \in \arg\min_{u \in \mathcal{F}} \mathcal{L}(u) = \mathbb{E}_{(X,Y)}[\log(1 + \exp^{-Yu(X)}]$.
- Consider ERM

$$\widehat{u} \in \arg\min_{u \in \mathcal{F}_{NN}} \widehat{\mathcal{L}}(u) = \sum_{i=1}^{n} \log(1 + \exp^{-Y_i u(X_i)})/n. \tag{2}$$

- Given an solver $\mathcal{A}$ that will output an $u_{\theta_{\mathcal{A}}} \in \mathcal{F}_{NN}$ s.t. $\widehat{\mathcal{L}}(u_{\theta_{\mathcal{A}}}) \leq \widehat{\mathcal{L}}(\widehat{u}) + \tau$.
- We want to bound $\mathcal{R}(u_{\theta_{\mathcal{A}}}) = \mathbb{E}_{\mathbb{D}}[\mathcal{L}(u_{\theta_{\mathcal{A}}}) - \mathcal{L}(u^*)]$ since

$$\mathbb{E}_{\mathbb{D}}[\mathbb{P}(\text{sign}(u_{\theta_{\mathcal{A}}}(X)) \neq Y)] \leq C\mathcal{R}^{1/2}(u_{\theta_{\mathcal{A}}}).$$

- $$\mathcal{R}(u_{\theta_{\mathcal{A}}}) \leq 2 \underbrace{\inf_{u \in \mathcal{F}_{NN}} \mathcal{L}(u) - \mathcal{L}(u^*)}_{\mathcal{E}_{\text{app}}} + \underbrace{\mathbb{E}_{\mathbb{D}}[\mathcal{L}(u^*) - 2\widehat{\mathcal{L}}(u_{\theta_{\mathcal{A}}}) + \mathcal{L}(u_{\theta_{\mathcal{A}}})]}_{\mathcal{E}_{\text{sta}}} + 2\underbrace{\tau}_{\mathcal{E}_{\text{opt}}}.$$

Complete error analysis

Approximation/Statistical error of FNN

- Approximation with composition.
  - Hilbert's 13 problem: Whether the solution of $x^7 + ax^3 + bx^2 + cx + 1 = 0$, $x(a, b, c)$, can be written as the composition of functions of only two variables.
  - Kolmogorov-Arnold superposition theorem 1957: $\forall u(x) \in C^0\left([0,1]^d\right), \exists \psi_{p,q}(x)$ and $\phi_q^u(x)$ defined on $\mathbb{R}$ such that $u(x) = \sum_{q=1}^{2d+1} \phi_q^u \left(\sum_{p=1}^{d} \psi_{p,q}\left(x_p\right)\right)$.
  - Prove the universal approximation of 2/3-layer NN with KAT Cybenko 1989, Kürková 1992.
- Recent development on deep approximation Telgarsky 2016, Yarotsky 2017, Petersen 2019, Zhou 2019, Yarotsky-Zhevnerchuk 2019, Shen-Yang-Zhang 2019-2023...
  - Suffers CoD, i.e., to approximate $u \in \mathcal{H}_\mu^\alpha([0,1]^d)$ to the error ϵ on need a NN u_θ with size $\mathcal{O}(C_d(1/\epsilon)^{d/\alpha} \log 1/\epsilon)$.
  - $\forall \epsilon > 0$, ReLU-sine-2^x network with the depth 6 and width $\max\{2d\lceil \log(\sqrt{d}(\frac{3\mu}{\epsilon})^{1/\alpha})\rceil, 2\lceil \log \frac{3\mu d^{\alpha/2}}{2\epsilon}\rceil + 2\}$ can approximate $f \in \mathcal{H}_\mu^\alpha([0,1]^d)$ **Jiao**-Lai-Lu-Wang-Yang ***SIMA*** 23.
- Statistical/ generalization error
  - Using tools in empirical process: symmetrization, contraction, Dudley's entropy integral, covering number, pseudodimension Ledoux-Talagrand 1991, Van Der Vaart-Wellner 1996, Bartlett-Harvey-Liaw-Mehrabian 2019.
    - $\mathcal{E}_{\text{sta}} \leq 4\text{Rad}(\ell_{\mathcal{F}}) := 4\mathbb{E}_{\mathbb{D},\tau}[\sup_f \left|\frac{1}{n}\sum_{i=1}^n \tau_i \ell(f, Z_i)\right|] \leq \tilde{\mathcal{O}}(\mathcal{W}^2\mathcal{D}^2/n)$.

Bias variance trade off

- Bias variance trade off ($\mathcal{E}_{\text{opt}} = 0$)
  - Setting $\mathcal{W}, \mathcal{D}$ to trade off the $\mathcal{E}_{\text{app}}$ and $\mathcal{E}_{\text{sta}}$ achieves minimax rate $\mathcal{O}(n^{-2\alpha/(2\alpha+d)})$ Bauer-Kohler 2020, Schmidt-Hieber 2021, Kohler-Krzyzak-Langer 2019, Chen-Jiang-Liao-Zhao 2019, Nakada-Imaizumi 2020, Farrell-Liang-Misra 2021, Fan-Gu 2022...
- Reduce the curse of dimensionality by the low-dim structure of the problems
  - Smoothness, Barron class, compositional structure, Low intrinsic dimension $d^* = \dim_M(\text{supp}(\mu_X)) \ll d$ **Jiao**-Wang-Yang ***ACHA*** 23, **Jiao**-Shen-Lin-Huang ***AoS*** 23.
- Error analysis in other type of learning tasks
  - Representation learning Chen-**Jiao**-Qiu-Yu ***AoS*** 24, Huang-**Jiao**-Liao-Liu-Yu ***TIT*** 24.
  - Regression Feng-**Jiao**-He-Kang-Wang ***JMLR*** 24, Ding-Duan-**Jiao**-Yang ***TIT*** 25.
  - Reinforcement learning Feng-**Jiao**-Kang-Zhang-Zhou ***JMLR*** 23, **Jiao**-Kang-Liu-Lu-Yang ***AoS*** 25.
  - Sampling and generative learning Huang-**Jiao**-Liu-Wang-Yang ***JMLR*** 22, Dai-**Jiao**-Kang-Lu-Yang ***SICON*** 23, Gao-Huang-**Jiao** ***JMLR*** 24, Huang-**Jiao**-Kang-Liao-Liu ***TIT*** 24, **Jiao**-Kang-Liu-Peng-Zuo ***TIT*** 25.
  - Learning with transformers **Jiao**-Lai-Sun-Wang-Yan arXiv:2504.12175, **Jiao**-Lai-Wang-Yan arXiv:2504.13558.

Complete error analysis

- Optimization error τ
  - Overparametrization makes neural network optimization easy, i.e., SGD with random initialization and small stepsize converges linearly since DNNs satisfy error bound like condition
$$\|\partial^L{}_\theta \widehat{\mathcal{L}}(u_\theta)\|^2 \geq c\widehat{\mathcal{L}}(u_\theta)$$
Polyak 63, Lojasiewicz 63, Jacot-Gabriel-Hongler 19, Allen-Li-Song 19, Du et al., 19, Nguyen-Pham 20, Nguyen 21, Liu-Zhu-Belkin 22.
  - Why overparametrized deep learning works is a mystery !!!
    - Double descent for linear and kennel models Bartlett-Long-Lugosi-Tsigler 19, Hastie-Montanari-Rosset-Tibshirani 2019, Liang-Rakhlin 2019, Nakkiran-Venkat-Kakade-Ma 2020, Belkin 21, Bartlett-Montanari-Rakhlin 21 Acta Numerica
    - Kohler-Krzyzak 2022 show that the estimation error in over-parameterized deep nonparametric least square regression is inconsistent.
- Complete error analysis $\|u_{\theta_{\mathcal{A}}} - u^*\|$ **Jiao**-Li-Wu-Yang-Zhang ***JMLR*** 25, **Jiao**-Lai-Wang ***TIT*** 25.

Projected gradient descent (PGD)

- PGD
  - Let $F(\boldsymbol{\theta}^{\mathfrak{m}}) = F\Big((\boldsymbol{\theta}^{\mathfrak{m}}_{in}, \boldsymbol{\theta}^{\mathfrak{m}}_{out})\Big) = \widehat{\mathcal{L}}(u_{\boldsymbol{\theta}^{\mathfrak{m}}})$.
  - $$\Big((\boldsymbol{\theta}^{\mathfrak{m}}_{in})^{[t+1]}, (\boldsymbol{\theta}^{\mathfrak{m}}_{out})^{[t+1]}\Big) = \text{Proj}_{A\times B}\bigg(\Big((\boldsymbol{\theta}^{\mathfrak{m}}_{in})^{[t]}, (\boldsymbol{\theta}^{\mathfrak{m}}_{out})^{[t]}\Big) - \lambda \nabla F\Big((\boldsymbol{\theta}^{\mathfrak{m}}_{in})^{[t]}, (\boldsymbol{\theta}^{\mathfrak{m}}_{out})^{[t]}\Big)\bigg),$$
- Initialization $(\boldsymbol{\theta}^{\mathfrak{m}})^{[0]}$
  Set
  $$(\boldsymbol{\theta}^{\mathfrak{m}}_{out})^{[0]} = 0,$$
  and the elements in $(\boldsymbol{\theta}^{\mathfrak{m}}_{in})^{[0]}$ as
  $$\big(a^{(\ell)}_{k,i,j}\big)^{[0]} \sim \text{Unif}[-B_{\boldsymbol{\theta}}, B_{\boldsymbol{\theta}}], \quad \big(b^{(\ell)}_{k,i}\big)^{[0]} \sim \text{Unif}[-B_{\boldsymbol{\theta}}, B_{\boldsymbol{\theta}}], \quad k = 1, ..., \mathfrak{m}, \ell = 1, ..., \mathcal{D}$$
- Constraint set
  Set
  $$A = \mathcal{B}_2((\boldsymbol{\theta}^{\mathfrak{m}}_{in})^{[0]}, \eta), \quad B = \mathcal{B}_1(0, M).$$
  - Exact algorithm for projection on ℓ_1 norm ball in linear time Duchi- Shwartz-Singer-Chandra 2008

Lemmas on (new) error decomposition

- Error decomposition
  - W.L.O.G, let $\bar{u} = \inf_{u \in \mathcal{PNN}} \|u - u^*\|^2_{H^1(\Omega)}$.

$$\|u_{\mathcal{A}} - u^*\|^2_{H^1(\Omega)} \leq C[\underbrace{\|\bar{u} - u^*\|^2_{H^1(\Omega)}}_{\mathcal{E}_{\text{app}}} + 2 \underbrace{\sup_{u \in \mathcal{PNN}} |\mathcal{L}(u) - \widehat{\mathcal{L}}(u)|}_{\mathcal{E}_{\text{sta}}} + \underbrace{\widehat{\mathcal{L}}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}(\bar{u})}_{\mathcal{E}^-_{\text{opt}}}]$$

    - $\mathcal{E}^-_{\text{opt}} \leq \mathcal{E}_{\text{opt}}$ implies a sharper error decomposition
    - Previous error decomposition decouple the total error to much
    - $\mathcal{E}^-_{\text{opt}}$ is easier to control than $\mathcal{E}_{\text{opt}}$ since we can use the property of $\bar{u}$
  - $$\mathcal{E}^-_{\text{opt}} \leq \underbrace{|\widehat{\mathcal{L}}(u_{\mathcal{A}}) - \widehat{\mathcal{L}}((u_{\widetilde{\theta}})^{[0]})|}_{(I) \quad \text{Err}_{\text{it}}} + \underbrace{|\widehat{\mathcal{L}}((u_{\widetilde{\theta}})^{[0]}) - \widehat{\mathcal{L}}(\bar{u})|}_{(II) \quad \text{Err}_{\text{ini}}}$$
    - Bound the initialization error via over-parametrization!

Main result

- **Theorem** Jiao-Li-Wu-Yang-Zhang ***JMLR*** 25
  - For any $0 < \epsilon \ll 1$, we use PGD algorithm for DRM training. If we set

$$\rho = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad \bar{\mathfrak{m}} = C_1(d)\epsilon^{-\frac{d}{2(1-\mu)}}, \quad M = C_2(d)\epsilon^{-\frac{3d}{4(1-\mu)}},$$

$$\mathcal{W} = 2^{\lceil \log_2(d+1) \rceil + 1}, \mathcal{D} = \lceil \log_2(d+1) \rceil + 2, B_\theta = C_3(d)\epsilon^{-1-\frac{d}{1-\mu}},$$

$$n = C_{\Omega,d}\epsilon^{-\frac{4(1-\mu+d)(\lceil \log_2(d+1) \rceil + 2)+3d}{1-\mu}-2}, \quad \eta = \epsilon^{-\frac{d}{1-\mu}}, \quad \delta = 1/n,$$

$$\bar{\epsilon} = C_{\Omega,d}\epsilon^{-\frac{3(1-\mu+d)(\lceil \log_2(d+1) \rceil + 2)+3d/2}{1-\mu}-1}, \quad R = C_{\Omega,d}\epsilon^{-\frac{(1-\mu+d)(3(\lceil \log_2(d+1) \rceil + 2)+1/2)+2d}{1-\mu}-1}\eta,$$

$$\mathfrak{m} = C_1(d)n\bar{\mathfrak{m}}(\frac{B_\theta}{\bar{\epsilon}})^{\mathcal{W}^2\mathcal{D}}, \quad T = C_{\Omega,d}\epsilon^{-\frac{2(1-\mu+d)(\lceil \log_2(d+1) \rceil + 2)+3d/2}{1-\mu}-1}\mathfrak{m}, \quad \lambda = 1/T.$$

Then, we have with probability at least $1 - 2 \cdot \epsilon^{\mathcal{O}(d/(1-\mu))}$

$$||u_{\theta_{\mathcal{A}}} - u^*||_{H^1} \leq \tilde{\mathcal{O}}(\epsilon),$$

where u^* solves (1), μ is an arbitrarily small positive number.

Sobolev-penalized regression

Sobolev-penalized regression

- Motivations: simultaneously recovering both the functions and their derivatives without derivative measurement.
  - Applications: $Y = f_0(X) + \epsilon$ with data $(X_i, Y_i)_{i=1,\dots,n}$ in score difference estimation, inverse problems and non-parametric variable selection...
- Difficulty: convergence in L^2-norm can not imply the convergence in H^1-norm.
- Solution: least-squares regression with gradient penalty

$$L^\lambda(f) := \underbrace{\mathbb{E}_{(X,Y)}\big[(Y - f(X))^2\big]}_{\text{least-squares}} + \underbrace{\lambda |f|^2_{H^1(\Omega)}}_{\text{gradient penalty}}.$$

  - Variational form: find $f^\lambda \in H^1(\Omega)$, such that for each $v \in H^1(\Omega)$,

$$(f^\lambda - f_0, v)_{L^2(\mu_X)} + \lambda(\nabla(f^\lambda - f_0), \nabla v)_{L^2(\Omega)} = \lambda(\nabla f_0, \nabla v)_{L^2(\Omega)}.$$

    - Considering $\mu_X = C_d \mathrm{d}x$ and let $v = f^\lambda - f_0$ implies

$$\overbrace{\|f^\lambda - f_0\|^2_{L^2(\Omega)}}^{\text{interior } L^2 \text{ error}} + \lambda \overbrace{|f^\lambda - f_0|^2_{H^1(\Omega)}}^{\text{interior gradient error}} \le \lambda \|\Delta f_0\|_{L^2(\Omega)} \|f^\lambda - f_0\|_{L^2(\Omega)}.$$

    - $$\|f^\lambda - f_0\|^2_{L^2(\Omega)} \le \lambda^2 \|\Delta f_0\|^2_{L^2(\Omega)}, \quad |f^\lambda - f_0|^2_{H^1(\Omega)} \le \lambda \|\Delta f_0\|^2_{L^2(\Omega)}.$$

Semi-supervised deep-Sobolev regression

- Empirical deep sobolev-penalized risk minimization

$$\widehat{f}^{\lambda}_{n,m} \in \underset{f \in \mathrm{conv}(\mathcal{F})}{\arg\min}\, \widehat{L}^{\lambda}_{n,m}(f) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - f(X_i))^2 + \frac{\lambda}{m}\sum_{j=1}^{m}\|\nabla f(Z_j)\|_2^2.$$

  where

  - Labeled data: $(X_i, Y_i) \sim^{\text{i.i.d.}} \mu_{X,Y}$.
  - Unlabeled data: $Z_j \sim^{\text{i.i.d.}} \mathrm{Unif}(\Omega)$.

- Assumptions
  - **A1. Sub-Gaussian noise.** The noise ϵ is sub-Gaussian with mean 0 and finite variance proxy σ^2.
  - **A2. Bounded hypothesis.** There exists an absolute positive constant B_0, such that $\sup_{x\in\Omega}|f_0(x)| \le B_0$. Further, functions in hypothesis class $\mathcal{F}$ are also bounded, that is, $\sup_{x\in\Omega}|f(x)| \le B_0$.
  - **A3. Bounded derivatives of hypothesis.** There exists positive constants $\{B_{1,k}\}_{k=1}^{d}$, such that $\sup_{x\in\Omega}|D_k f_0(x)| \le B_{1,k}$ for $1 \le k \le d$. Further, the first-order partial derivatives of functions in hypothesis class $\mathcal{F}$ are also bounded, i.e., $\sup_{x\in\Omega}|D_k f(x)| \le B_{1,k}$ for each $1 \le k \le d$ and $f \in \mathcal{F}$. Denote by $B_1^2 := \sum_{k=1}^{d} B_{1,k}^2$.
  - **A4. Regularity of regression function.** The regression function satisfies $\Delta f_0 \in L^2(\Omega)$ and $\nabla f_0 \cdot \mathbf{n} = 0$ a.e. on $\partial\Omega$, where $\mathbf{n}$ is the unit normal to the boundary.

Error Decomposition

- Oracle inequality

  Under **A1** to **A4**. Suppose $n \geq \log N(B_0\delta, \mathcal{F}, L^2(\mathcal{D}))$ and $m \geq \max_k \log N(B_{1,k}\delta, D_k\mathcal{F}, L^2(\mathcal{S}))$. Then

$$\mathbb{E}\big[\|\widehat{f}^{\lambda}_{n,m} - f_0\|^2_{L^2(\Omega)}\big] \lesssim \beta\lambda^2 + \varepsilon_{\text{app}}(\mathcal{F}, \lambda) + \varepsilon_{\text{sta}}(\mathcal{F}, n) + \varepsilon^{\text{reg}}_{\text{sta}}(\nabla\mathcal{F}, m),$$

$$\mathbb{E}\big[\|\nabla(\widehat{f}^{\lambda}_{n,m} - f_0)\|^2_{L^2(\Omega)}\big] \lesssim \beta\lambda + \lambda^{-1}\varepsilon_{\text{app}}(\mathcal{F}, \lambda) + \lambda^{-1}\varepsilon_{\text{sta}}(\mathcal{F}, n) + \lambda^{-1}\varepsilon^{\text{reg}}_{\text{sta}}(\nabla\mathcal{F}, m),$$

  where

  - $\beta = \|\Delta f_0\|^2_{L^2(\Omega)} + B_1^2$.

  - The approximation error $\varepsilon_{\text{app}}(\mathcal{F}, \lambda)$, the statistical errors $\varepsilon_{\text{sta}}(\mathcal{F}, n)$ and $\varepsilon^{\text{reg}}_{\text{sta}}(\nabla\mathcal{F}, m)$ are defined as

$$\varepsilon_{\text{app}}(\mathcal{F}, \lambda) = \inf_{f\in\mathcal{F}} \Big\{\|f - f_0\|^2_{L^2(\Omega)} + \lambda\|\nabla(f - f_0)\|^2_{L^2(\Omega)}\Big\},$$

$$\varepsilon_{\text{sta}}(\mathcal{F}, n) = (B_0^2 + \sigma^2)(\log n) \inf_{\delta>0} \Big\{\Big(\frac{2\log N(B_0\delta, \mathcal{F}, L^2(\mathcal{D}))}{n}\Big)^{\frac{1}{2}} + \delta\Big\},$$

$$\varepsilon^{\text{reg}}_{\text{sta}}(\nabla\mathcal{F}, m) = B_1^2 \inf_{\delta>0} \Big\{\max_{1\leq k\leq d} \frac{\log N(B_{1,k}\delta, D_k\mathcal{F}, L^2(\mathcal{S}))}{m} + \delta\Big\}.$$

Convergence Rate

- Let $\Omega \subseteq K \subseteq \mathbb{R}^d$ be two bounded domain. Set the hypothesis class as a deep ReQU neural network $\mathcal{F} = \mathcal{N}(L, W, S)$ with $L = \mathcal{O}(\log N)$ and $S = \mathcal{O}(N^d)$. Then for each $\phi \in C^s(K)$ with $s \in \mathbb{N}_{\geq 2}$, there exists $f \in \mathcal{F}$ such that

$$\|f - \phi\|_{L^2(\Omega)} \leq CN^{-s}\|\phi\|_{C^s(K)}, \quad \|\nabla(f - \phi)\|_{L^2(\Omega)} \leq CN^{-(s-1)}\|\phi\|_{C^s(K)},$$

where C is a constant independent of N.

- Suppose the activation function is piecewise-polynomial. Let $\mathcal{D} = \{X_i\}_{i=1}^n$ and $\mathcal{S} = \{Z_j\}_{j=1}^m$. Then

$$\log N(\delta, N(L, S, B), L^2(\mathcal{D})) \lesssim LS \log(S) \log\left(\frac{nB}{\delta}\right),$$
$$\log N(\delta, D_k N(L, S, B), L^2(\mathcal{S})) \lesssim L^2 S \log(S) \log\left(\frac{mB}{\delta}\right).$$

- Theorem Ding-Duan-**Jiao**-Yang ***TIT*** 25
  Under **A1** to **A4**. Let $\Omega \subseteq K \subseteq \mathbb{R}^d$ be two bounded domain. Assume that $f_0 \in C^s(K)$ with $s \in \mathbb{N}_{\geq 2}$. Set the hypothesis class as a deep ReQU neural network class $\mathcal{F} = N(L, W, S)$ with $L = \mathcal{O}(\log n)$, $S = \mathcal{O}(n^{\frac{d}{d+4s}})$ and Let $\lambda = \mathcal{O}(n^{-\frac{s}{d+4s}} \log^2 n)$. Then

$$\mathbb{E}\left[\|\widehat{f}^{\lambda}_{n,m} - f_0\|^2_{L^2(\Omega)}\right] \leq \widetilde{\mathcal{O}}\left(n^{-\frac{2s}{d+2s+2s}}\right) + \widetilde{\mathcal{O}}\left(n^{\frac{d}{d+4s}}/m\right),$$
$$\mathbb{E}\left[\|\nabla(\widehat{f}^{\lambda}_{n,m} - f_0)\|^2_{L^2(\Omega)}\right] \leq \widetilde{\mathcal{O}}\left(n^{-\frac{2(s-1)-(s-2)}{d+2s+2s}}\right) + \widetilde{\mathcal{O}}\left(n^{\frac{d+s}{d+4s}}/m\right).$$

Application 1: nonparametric variable selection

- **A5. Sparsity structure** There exists $f_0^* : \mathbb{R}^{d^*} \to \mathbb{R}$ ($1 \leq d^* < d$) such that for each $x := (x_1, \ldots, x_d) \in \mathbb{R}^d$,

$$f_0(x_1, \ldots, x_d) = f_0^*(x_{j_1}, \ldots, x_{j_{d^*}}), \quad \{j_1, \ldots, j_{d^*}\} \subseteq [d].$$

- The derivatives can indicate whether a variable is relevant to the output.
  - A variable $k \in [d]$ is irrelevant for the function f with respect to Lebesgue measure on Ω, if

$$D_k f(X) = 0 \quad \text{almost surely},$$

  and relevant otherwise. The set of relevant variables is defined as

$$\mathcal{I}(f) = \{k \in [d] : \|D_k f\|_{L^2(\Omega)} > 0\}.$$

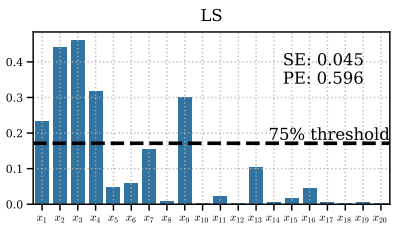
- Under **A5**, the convergence rate improved by replacing d with d^*.

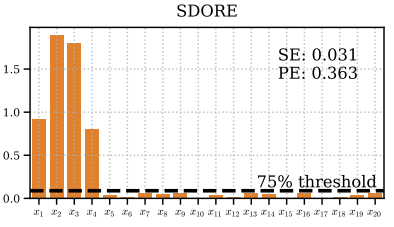
- Selection consistency: under **A1** to **A5**, $\lim_{n\to\infty} \Pr\left\{\mathcal{I}(f_0) = \mathcal{I}(\widehat{f}^{\lambda}_{n,m})\right\} = 1$, where $\lambda = \mathcal{O}(n^{-\frac{s}{d^*+4s}} \log^2 n)$, and m is sufficiently large.

Numerics 1

$$f_0(x_1, \ldots, x_{20}) = \sum_{i=1}^{3} \sum_{j=i+1}^{4} x_i x_j.$$

sparse structure





Application 2: inverse source problem

- Elliptic equation with unknown source

$$\begin{cases} -\nabla \cdot (a(x)\nabla u(x)) + c(x)u = f_0(x), & \text{in } \Omega, \\ \nabla u \cdot \mathbf{n} = 0, & \text{on } \partial\Omega. \end{cases}$$

  - Measurement model

$$Y = \mathcal{S}(f_0)(X) + \epsilon,$$

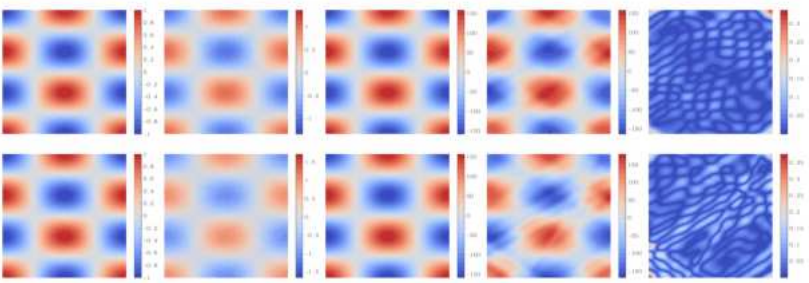
    where $u_0 = \mathcal{S}(f_0)$ is the solution map.
  - We have interior position-measurement pairs: $\{(X_i, Y_i)\}_{i=1}^n$ (expensive to get). And positions measurement $Z_1, \ldots, Z_m \sim^{\text{i.i.d.}} \text{Unif}(\Omega)$ (cheap to get).

- Recovering procedure
  - Train $\widehat{u}^{\lambda}_{n,m}$.
  - Recover $\widehat{f}^{\lambda}_{n,m} = -\nabla \cdot (a(x)\nabla \widehat{u}^{\lambda}_{n,m}) + c(x)\nabla \widehat{u}^{\lambda}_{n,m}$.

- Convergence rate in weak norm

$$\mathbb{E}\left[\|\widehat{f}^{\lambda}_{n,m} - f_0\|_{(H^1(\Omega))^*} \right] \lesssim \mathcal{O}\left(n^{-\frac{s}{2(d+4s)}} \right).$$

Numerics 2



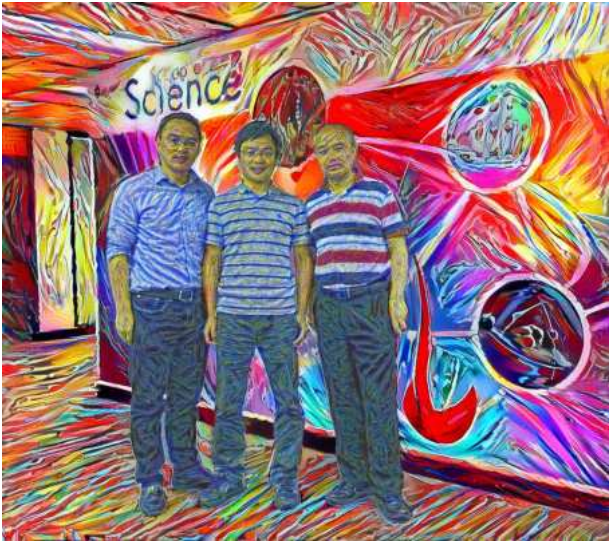
(a) Clear data (b) Noisy data (c) Exact source (d) Recovery (e) Point-wise error

Outline

Domain adaptation via GAN

Domain adaptation via GAN

Domain adaptation via GAN

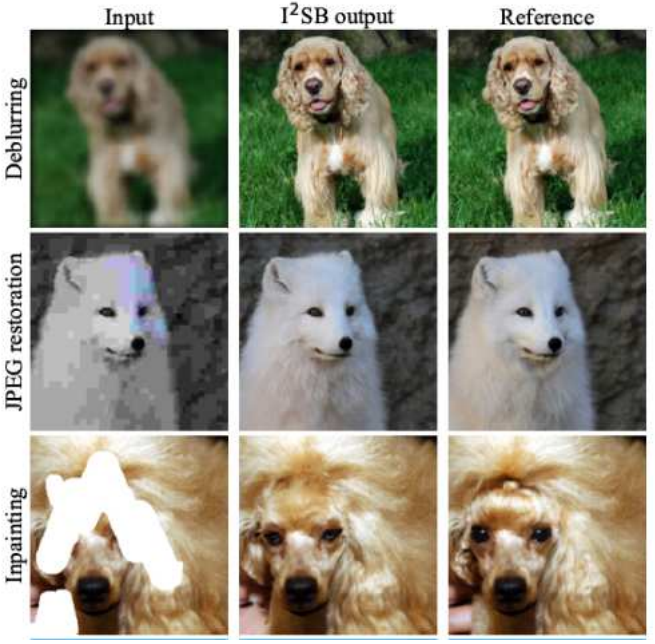


Based on SycleGAN

Image restoration via diffusion model

"Supperresolution, $128 \times 128 \rightarrow 512 \times 512$"

Image restoration via Schrödinger-Bridge



"Image restoration via I^2SB (Liu-Vahdat-Huang-Theodorou-Anandkumar 2023)"

Image generation via diffusion model

"Rushing water plunges three thousand feet, supremely beautiful"

Image generation via stable diffusion model

”Small bridge, flowing water, people’s homes, Van Gogh style”

Generative learning

- Generative learning (nonparametric density estimation), learn ν in $\mathbb{R}^d$ from data $X_1, \ldots, X_n$.
  - An important and challenging problem in statistics, machine learning is learn the underlying high-dimension distribution ν or its density function $p(\cdot)$ from data.
- Conditional generative learning.
  - Controlabel generation.
    - AGI model such as Stable Diffusion, Sora.
  - More than prediction.
    - Learn the conditional distribution (quanlify the uncertanty) $\nu(\cdot|X = x)$ from paired data (X_1, Y_1), ..., (X_n, Y_n) rather than the conditional mean $\mathbb{E}[Y|X = x]$.

Deep generative learning

- Deep generative learning: sample $Z \sim \mu$ and learn a map g such that $g_{\#}\mu \approx \nu$.
  - VAE Kingma-Welling 2014 $\approx$ probabilistic deep nonlinear factor model $+$ variational EM $+$ sampling.
  - GAN Goodfellow et al. 2014 $\approx$ deep nonlinear factor analysis $+$ training with sequential differentiable two sample test.
    - GAN cited by 83K. GANs is "the coolest idea in deep learning in the last 10 years." (Yann LeCunn)
  - SDE based methods deeply rooted in inexact Langevin diffusion, Schrödinger Bridge, Nelson's stochastic mechanics.
    - Generative learning via diffusion model in latent space is the key of Stable Diffusion Robin-Andreas-Dominik-Patrick-Björn 22 and Sora https://openai.com/research/video-generation-models-as-world-simulators.

Motivation and road maps

- Existing methods
  - Inverse transform method.
  - Metropolis-Hastings (MH) algorithm Metropolis-Rosenbluth-Teller-Teller 1953, Hastings, 1970.
  - Gibbs sampler Geman-Geman 1984, Gelfand-Smith 1990.
  - Langevin sampler Arnold 1974, Roberts-Tweedie 1996; Containt version Chang-Tang-Zhu 23.
  - Hamiltonian Monte Carlo (HMC) Duane-Kennedy-Pendleton-Roweth 1987, Neal 2011.
  - See the review paper Changye-Robert 2020, Dunson and Johndrow 2020 ...
- Motivations
  - Meta stability
    - For HMC sampler on 1-dimensional mixture $0.5\mathcal{N}(-1,\sigma^2)+0.5\mathcal{N}(1,\sigma^2)$, the mixing time is bounded by $\exp(1/(2\sigma^2))$.
  - The curse of dimensionally
    - For Langevin sampler, the mixing time is bounded by $\mathcal{O}(\frac{d}{\epsilon}\log(d/\epsilon)C^{5/2})$ under the assumption $\mu\in \mathrm{LogSob}(C)$.

      Durmus-Moulines 17, Eberle-Guillin-Zimmer 17, Cattaux-Guillin-Monmarce-Zhang 17,

      Mou-Flammarion-Wainwrighty-Bartlett 19, Bortoli-Durmus 19.
    - $C=\mathcal{O}(\exp^d)$ Menz-Schlichting 14.
  - The difficulties are resulted from the ergodicity requirements.
    - Sampling without the ergodicity via Schrödinger Bridge

      Huang-**Jiao**-Kang-Liao-Liu ***TIT*** 24, Dai-**Jiao**-Kang-Lu-Yang ***SICON*** 23
  - Develop a sampling method like inverse transform method Feng-Gao-Huang-**Jiao**-Liu ***JCGS*** 25 with theoretical guarantee Ding-**Jiao**-Lu-Yang-Yuan arXiv:2311.03660.

Motivations and road maps

- GANs.
  - Takehome massage: the model of GANs is nice Liang 21, Huang-**Jiao**-Li-Liu-Wang-Yang ***JMLR*** 22, **Jiao**-Wang-Yang ***ACHA*** 23, Chakraborty-Bartlett 24 and the Instability of training of GANs is due to algorithms.
- SDE based generative model.
  - Takehome message: SDE based models are stable to train but expensive (compared with GANs) Wang-**Jiao**-Xu-Wang-Yang ***ICML*** 21, Chen-Huang-Zhao-Wang 23, Oko-Akiyama-Suzuki 23, **Jiao**-Kang-Liu-Lin-Zuo arXiv:2404.13309.
- Generative learning via ODE flows.
  - Methods induced by gradient flow in probability measure spaces (ODE flows) can do one step generation Gao-**Jiao**-Wang-Wang-Yang-Zhang ***ICML*** 19, Gao-Huang-**Jiao**-Liu-Lu-Yang ***MSML*** 21, Song-Dhariwal-Chen-utskever 23, but the analysis is incomplete.
  - Takehome massage: we develop a stable and provable one step generation method based on ODE flow
    - establish end-2-end analysis of generative model (both conditional and unconditional) based on ODE flows Gao-Huang-**Jiao** ***JMLR*** 24, Chang-Ding-**Jiao**-Li-Yang arXiv:2402.01460, Gao-Huang-**Jiao**-Zheng arXiv:2404.00551.
    - propose and analyze characteristic learning for one step generation Ding-Duan-**Jiao**-Li-Yang-Zhang arXiv:2405.05512.

Error of GAN

GANs with IPM

- Define the integral probability metric (IPM) Muller 1997 with respect to the discriminator class $\mathcal{F}$:

$$d_{\mathcal{F}}(\nu, g_{\#}\mu) := \sup_{f\in\mathcal{F}} \mathbb{E}_{\nu}[f] - \mathbb{E}_{g_{\#}\mu}[f] = \sup_{f\in\mathcal{F}} \mathbb{E}_{X\sim\nu}[f(X)] - \mathbb{E}_{Z\sim\mu}[f(g(Z))].$$

  - when $\mathcal{F} = \mathrm{Lip}(1)$, $d_{\mathcal{F}} = \mathcal{W}_1$ is used in Wasserstein GAN Arjovsky et al., 2017.
  - when $\mathcal{F}$ is a Sobolev function class, $d_{\mathcal{F}}$ is used in Sobolev GAN Mroueh et al., 2018.

- GAN loss at the population level

$$\min_{g\in\mathcal{G}} d_{\mathcal{F}}(\nu, g_{\#}\mu) = \min_{g\in\mathcal{G}} \max_{f\in\mathcal{F}} \mathbb{E}_{X\sim\nu}[f(X)] - \mathbb{E}_{Z\sim\mu}[f(g(Z))].$$

- GAN loss at the sample level

$$(\hat{g}, \hat{f}) \in \arg \min_{g_\theta\in\mathcal{G}_{NN}} \max_{f_\omega\in\mathcal{F}_{NN}} \frac{1}{n}\sum_{j=1}^{n} f_\omega(X_j) - \frac{1}{m}\sum_{i=1}^{m} f_\omega(g_\theta(Z_i)).$$

- We want to bound $\mathbb{E}[d_{\mathcal{F}}(\nu, \hat{g}_{\#}\mu)]$ with $\mathcal{F} = \mathcal{H} = \mathcal{H}^{\beta}(\mathbb{R}^d)$.
  - On $[0,1]^d$ and $\beta = 1$, $\mathcal{W}_1(\nu, \hat{g}_{\#}\mu) \le \sqrt{d} d_{\mathcal{H}^1}(\nu, \hat{g}_{\#}\mu))$.

Main results

- **Theorem** Huang-Jiao-Li-Liu-Wang-Yang ***JMLR*** 22

  Assume
  - The reference distribution μ on $\mathbb{R}^k$ with $k < d$ is absolutely continuous with respect to the Lebesgue measure.
  - The target ν is supported on $[0,1]^d$.

  Set
  - $\mathcal{G}_{NN} = \{g \in \mathcal{NN}(W_1, L_1) : g(\mathbb{R}^k) \subseteq [0,1]^d\}$ with $W_1^2 L_1 \asymp n$,
  - $\mathcal{F}_{NN} = \mathcal{NN}(W_2, L_2) \cap \mathrm{Lip}(\mathbb{R}^d, K, 1)$ with
    $W_2 L_2 \precsim n^{1/2} \log^2 n, \quad K \precsim (W_2 L_2)^{2+\max\{(4\beta-4)/d, 0\}} L_2 2^{L_2^2}$,
  - $m \gtrsim n^{2+2\beta/d} \log^6 n$.

  Then
  $$\mathbb{E}[d_{\mathcal{F}}(\nu, \hat{g}_{\#}\mu)] \precsim n^{-\beta/d} \vee n^{-1/2} \log n.$$

- Remarks
  - $\mathcal{O}(n^{-\beta/d})$ is minimax optimal under $d_{\mathcal{F}}$ for ν without density Liang 21, Singh et al. 18.
  - improve to $\mathcal{O}(n^{-\beta/d^*} + \sigma^2)$ if $\nu = \tilde{\nu} \otimes \xi$ where $\dim_M(\mathrm{supp}(\tilde{\nu})) = d^* < d$, ξ has zero mean and bounded variance σ^2.
  - improve to optimal rate $\mathcal{O}(n^{-(\beta+\alpha)/d} \vee n^{-1/2} \log n)$ if ν has density $p_\nu \in \mathcal{H}^\alpha([0,1]^d)$ (using regularized $\hat{\nu}$).
  - extend to target with unbounded supports but subexponential tail.

Sketch of the proof and extensions

- Sketch of the proof

$$d_{\mathcal{F}}(\nu, \hat{g}_{\#}\mu) \leq 4\mathcal{E}_1 + \mathcal{E}_2 + \mathcal{E}_3 + 2\mathcal{E}_4$$

with

  - $\mathcal{E}_1 = \mathcal{E}(\mathcal{F}, \mathcal{F}_{NN})$ the function approximation error of the discriminator class $\mathcal{F}_{NN}$ to $\mathcal{F}$ with Lipschitz control. Based on recent work of Daubechies-DeVore-Foucart-Hanin-Petrova 2019, Shen-Yang-Zhang 2021, Lu-Shen-Yang-Zhang 2021

  - $\mathcal{E}_2 = \inf_{g \in \mathcal{G}_{NN}} \|g_\sharp \mu - \hat{\nu}\|_{\mathcal{F}_{NN}}$ the distribution approximation error of the generator class $\mathcal{G}_{NN}$ (contruted by Yunfei).

  - $\mathcal{E}_3 = \|\hat{\nu}_- \nu\|_{\mathcal{F}}$ and $\mathcal{E}_4 = \|\hat{\mu} - \mu\|_{\mathcal{F}_{NN} \circ \mathcal{G}_{NN}}$ are the empirical process under the discrinator class $\mathcal{F}$ and or under the neural nets evaluation class $\mathcal{F}_{NN} \circ \mathcal{G}_{NN}$, respectively. Based on Dudley 1967, Van der Vaart-Wellner 1996, Schreuder 2020, Bartlett-Harvey-Liaw-Mehrabian 2019

- Extention to norm controlled network class which allows overparametrized generator and discrimator **Jiao**-Wang-Yang ***ACHA*** 23

- Extention to conditional GANs Zhou-**Jiao**-Liu-Huang ***JASA*** 23 and Bi-GANs Liu-Yang-Huang-**Jiao**-Wang ***NeruIPS*** 21

Gradient flow

Key idear

- The key idea is to solving in the primal form (GAN solving in the dual form) !
  - Give density q we want to find a transport map T such that the pushforward $T_\sharp q$ approximate the target p better than q.
  - Let $x \sim q$, if T is invertible, distribution of $T(x)$ is $T_\sharp q(x) = q(T^{-1}(x))|\det(\nabla_x T^{-1}(x))|$.
  - Set $T : \mathbb{R}^d \mapsto \mathbb{R}^d$ as a perturbation of identity map along the direction of $\mathbf{v} \in [L^2(q)]^d$, i.e, $Tx = x + t\mathbf{v}(x)$.
  - When t is small, the above T is invertible, change q gradually, easy to compute.
  - Define $\mathcal{L}(\mathbf{v}) = \mathbb{D}_{\mathrm{KL}}((\mathbb{I} + \mathbf{v})_\sharp q||p) : [L^2(q)]^d \mapsto \mathbb{R}^1$, we want to find $\mathcal{L}(\mathbf{v}) \leq \mathcal{L}(\mathbf{0})$.
- First order variation If $\lim_{\|x\|_2 \to \infty} \|\mathbf{g}(x)q(x)\| = 0$,

$$\langle -\partial \mathcal{L}_q(\mathbf{0}), \mathbf{g} \rangle_{L^2(q)} = \langle \nabla \log \frac{p}{q}, \mathbf{g} \rangle_{L^2(q)}.$$

Calculation of variation

- We want to find $\mathbf{v} \in [L^2(q)]^d$ that most decreases $\mathcal{L}(\mathbf{0})$, i.e., $\mathbf{v} = -\partial\mathcal{L}(\mathbf{0})$.
  - The first order variation is defined as

$$\langle -\partial\mathcal{L}(\mathbf{0}), \mathbf{g}\rangle_{L^2(q)} = -\nabla_t \mathcal{L}(t\mathbf{g})|_{t=0}, \quad \forall g \in [L^2(q)]^d.$$

  - Given $x \sim p$, consider the density of $T^{-1}x$ denoted as the pullback density $T^\sharp p(x) = p(T(x))|\det(\nabla_x T(x))|$.
  - Observing for small t

$$\log p(x + t\mathbf{g}(x)) = \log p(x) + t\mathbf{g}(x)^T \nabla_x \log p(x) + \mathcal{O}(t^2),$$
$$\log\det(I + t\nabla_x \mathbf{g}(x)) = t\,\mathrm{trace}\,(\nabla_x \mathbf{g}(x)) + \mathcal{O}(t^2).$$

$$\nabla_t \mathbb{D}_{\mathrm{KL}}((\mathbb{I} + t\mathbf{g})_\sharp q||p) = \nabla_t \mathbb{D}_{\mathrm{KL}}(q||(\mathbb{I} + t\mathbf{g})^\sharp p) = -\nabla_t \mathbb{E}_{x\sim q} \log[(\mathbb{I} + t\mathbf{g})^\sharp p(x)]dx$$
$$= -\mathbb{E}_{x\sim q}\nabla_t[\log p(x + t\mathbf{g}(x)) + \log|\det(I + t\nabla_x \mathbf{g}(x))|]dx.$$

  - We deduce $-\nabla_t \mathcal{L}_q(t\mathbf{g})|_{t=0} = \mathbb{E}_{x\sim q}[\langle \mathbf{g}(x), \nabla_x \log p(x)\rangle + \mathrm{trace}\,(\nabla_x \mathbf{g}(x))] = \mathbb{E}_{x\sim q}[\langle \nabla \log \frac{p(x)}{q(x)}, \mathbf{g}(x)\rangle]$ as long as $\lim_{\|x\|_2\to\infty} \|\mathbf{g}(x)q(x)\| = 0$.

Deep density-ratio estimation and algorithm

- Let $r^*(x) = p(x)/q(x)$ be the density ratio, then

$$r^* \in \arg\min \mathcal{L}(r) = \mathbb{E}_{X\sim q}[r(X)^2] - 2\mathbb{E}_{X\sim p}[r(X)] + 1.$$

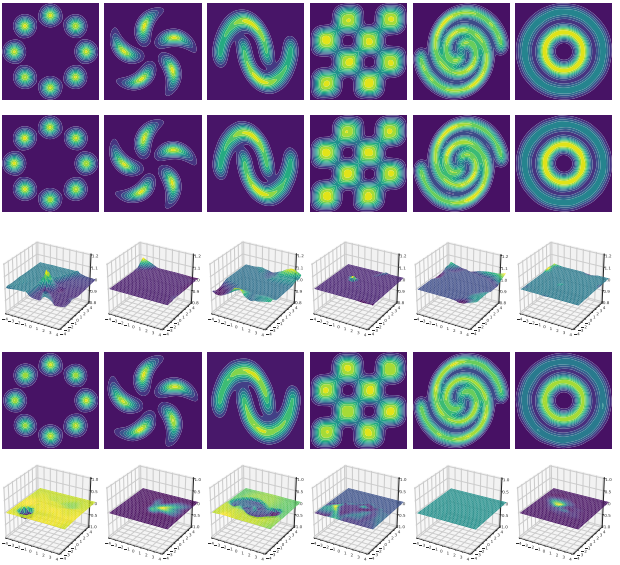
- Suppose $\{X_i\}_{i=1}^n$ and $\{Y_i\}_{i=1}^n$ are two collections of i.i.d data from densities $p(x)$ and $q(x)$, respectively. Considering

$$\widehat{r}_\phi \in \arg\min_{r_\phi \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}[r_\phi(Y_i)^2 - 2r_\phi(X_i)]. \tag{3}$$

Algorithm (inexact Riemannian gradient descent) Gao-**Jiao**-Wang-Wang-Yang-Zhang ***ICML*** 19, Gao-Huang-**Jiao**-Liu-Lu-Yang ***MSML*** 21

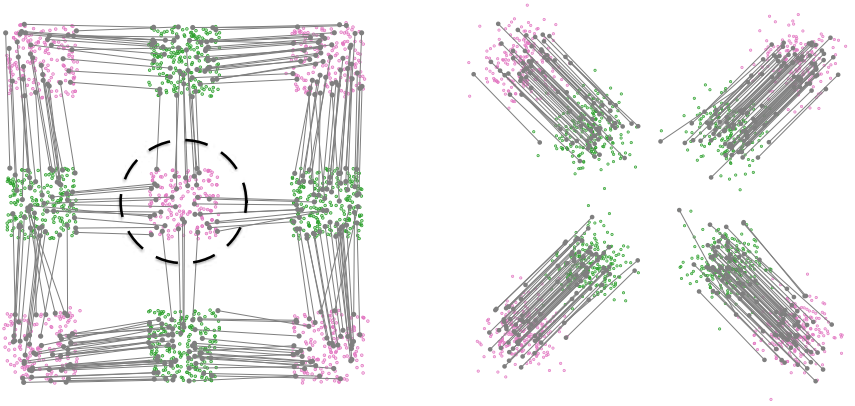
- Input $X_1, ...X_n$ sampled from p, $Y_1, ..., Y_n$ sampled from a initial reference via $Y_i = G_\theta(Z_i)$.
- Do the following iteratively
  - Get $\hat{\mathbf{v}}(x) = \nabla \log(\widehat{r}_\phi(x))$ via solving (3). Set $\widehat{T} = \mathbb{I} + s\hat{\mathbf{v}}$ with a small step size s.
  - Update the samples $Y_i = \widehat{T}(Y_i)$, $i = 1, \ldots, n$.
- Train the generator $G_\theta(\cdot)$ via solving $\min_\theta \sum_{i=1}^n \|G_\theta(Z_i) - Y_i\|_2^2/n$.

Numerical experiments: 2-D distributions



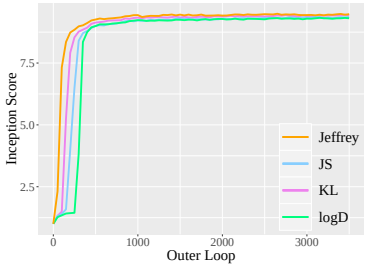
KDE plots of the target samples (the first row) and the corresponding generated samples (the second row and the fourth row). The third/five row shows surface plots of estimated density ratio (difference) after 20k iterations.

Numerical experiments: 2-D distributions

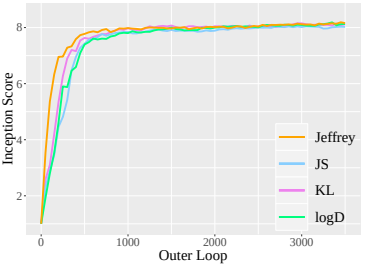


Learned transport maps between 5*squares* from 4*squares*, and *large*4*gaussians* from *small*4*gaussians*.

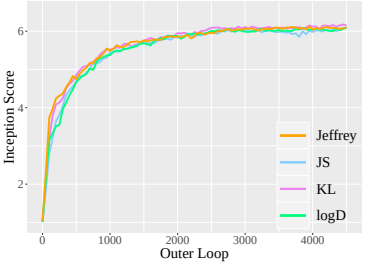
Stability



(a) MNIST

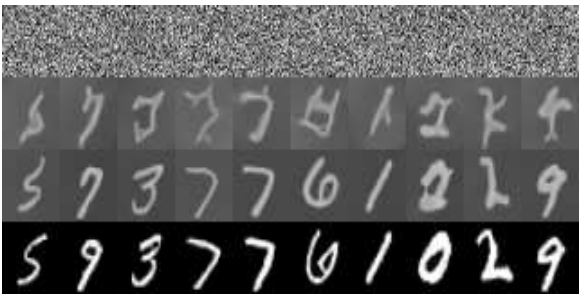


(b) FashionMNIST



(c) CIFAR10

Particle evolution of EPT on MNIST

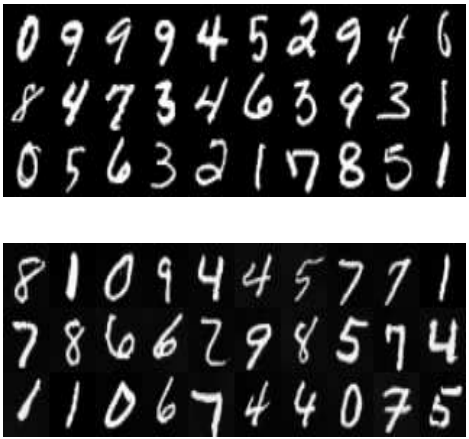


Particle evolution of EPT on MNIST (in $\mathbb{R}^{2352}$)

Particle evolution of EPT on CIFAR10

Particle evolution of EPT on CIFAR10 (in $\mathbb{R}^{3072}$).

Numerical experiments: visual comparisons



Visual comparisons between real image (top) and generated image (bottom) of MNIST

Numerical experiments: visual comparisons

Visual comparisons between real image (top) and generated image (bottom) of CIFAR10

Numerical experiments: visual comparisons

Visual comparisons between real image (top) and generated image (bottom) of CelebA

Gradient flow interpretation

- Let $q_t(x)$ be the density of x_t, $t \geq 0$. Consider $x_{t+\mathrm{d}t} = x_t + \mathbf{v}_t(x_t)\mathrm{d}t$ with infinitesimal $\mathrm{d}t$ and $\mathbf{v}_t = \nabla \log(p/q_t)$. Then

$$\frac{\partial}{\partial t} q_t = -\nabla \cdot (q_t \mathbf{v}_t) \ \text{ in } \ \mathbb{R}^+ \times \mathbb{R}^d \ \text{ with } \ q_0 = q, \tag{4}$$

  - Let $T(x) = x + \mathbf{v}_t(x)\mathrm{d}t$. $\log q_{t+\mathrm{d}t}(x) - \log q_t(x) = \frac{q_{t+\mathrm{d}t}(x) - q_t(x)}{q_t(x)} + o(\mathrm{d}t)$ and $\log q_{t+\mathrm{d}t}(x) = \log q_t(T^{-1}(x)) + \log|\det(\nabla_x T^{-1})| =$ $\log q_t(x - \mathbf{v}_t(x)\mathrm{d}t) + \log|\det(\nabla_x(x - \mathbf{v}_t(x)\mathrm{d}t))| + o(\mathrm{d}t) =$ $\log q_t(x) - \mathbf{v}_t(x)^T \nabla_x \log q_t(x)\mathrm{d}t - \nabla \cdot \mathbf{v}_t(x)\mathrm{d}t + o(\mathrm{d}t)$, implies $q_{t+\mathrm{d}t}(x) - q_t(x) =$ $q_t(x)(\log q_{t+\mathrm{d}t}(x) - \log q_t(x)) + o(\mathrm{d}t) = -\nabla \cdot (\mathbf{v}_t(x) q_t(x))\mathrm{d}t + o(\mathrm{d}t)$.
- The particle form of (4) is

$$\frac{\mathrm{d}}{\mathrm{d}t} x_t = \mathbf{v}_t(x_t), \ t > 0, \ \text{with } \ x_0 \sim q. \tag{5}$$

- We use KL-divergence to measure the discrepancies, i.e., $\mathcal{L}[q_t] = \mathbb{D}_{\mathrm{KL}}(q_t \| p)$.
  - Using Otto calculus on $(\mathcal{P}_2(\mathbb{R}^m), \mathcal{W}_2)$ (Riemannian structure) Otto 2001, Villani 2008
    - $$\frac{\partial}{\partial t} q_t = -\nabla^{\mathcal{W}} \mathcal{L}[q_t] = -\nabla \cdot (q_t \mathbf{v}_t), \ \ q_0 = q$$

      with $\mathbf{v}_t(x) = \nabla(-\frac{\partial \mathcal{L}[q_t]}{\partial q_t}(x)) = \nabla \log r_t(x) = \nabla \log \frac{p(x)}{q_t(x)}$.
    - $\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}[q_t] \leq 0$. If $p \in \mathrm{LogSob}(c)$, $\mathcal{L}[q_t] = \mathcal{O}(\exp^{-ct})$ Otto-Villani 2000
- EPT can be seen as inexact Riemannian gradient descent on measure spaces:

$$q_{k+1} = \exp_{q_k}^{s\mathbf{v}_k} \approx \exp_{q_k}^{s\hat{\mathbf{v}}_k} \approx (\mathbb{I} + s\hat{\mathbf{v}}_k)_{\#} q_k.$$

Inexact Langevian and diffusion model

Langevin diffusion

- The continuation equation (4) can be reformed as

$$\frac{\partial}{\partial t} q_t = -\nabla \cdot (q_t \nabla \log p) + \triangle q_t \ \text{ in } \ \mathbb{R}^+ \times \mathbb{R}^d \ \text{ with } \ q_0 = q, \tag{6}$$

- Let $p(x) = \exp^{-U(x)} / Z$ where ∇U is Lipschitz.
  - $\Pi = p(x)\mathrm{d}x$ is be the unique invariant probability distribution of Arnold 74 of the Langevin SDE
$$\mathrm{d}X_t = -\nabla U(X_t)\,\mathrm{d}t + \sqrt{2}\mathrm{d}B_t.$$
- Unadjusted Langevin Algorithm (ULA) based on Euler-Maruyama (EM) Dalalyan 17, Durmus-Moulines 17, Eberle-Guillin-Zimmer 17, Cattaux-Guillin-Monmarce-Zhang 17, Mou-Flammarion-Wainwrighty-Bartlett 19, Bortoli-Durmus 19....
  - $X_{k+1} = X_k - \gamma_k \nabla U(X_k) + \sqrt{2\gamma_k}\xi_k$, ξ_k are i.i.d. standard Gaussian, γ_k are step size.
- Generative learning via Langevin diffusion
  Given i.i.d data $X_1, ...X_n \sim p(x) = \exp^{-U(x)} / Z$ with $U(x)$ unknown, we want to learn p implicitly via sampling.

Langevin for generative learning with score matching

- Recall we can sample data from $p(x)$ with unadjusted Langevin algorithm (ULA) $X_{k+1} = X_k - \gamma_k \nabla U(X_k) + \sqrt{2\gamma_k}\xi_k$, if we have $-\nabla U(x) = \nabla \log p(x)$.

- First estimate the score $s(x) = \nabla \log p(x)$ using deep neural network, say $\hat{s}_\theta(x)$. Then sample via $X_{k+1} = X_k + \gamma_k \hat{s}_\theta(X_k) + \sqrt{2\gamma_k}\xi_k$ Song-Ermon 19.

  - $$v^\sigma(x) \in \arg\min_v \mathbb{E}_{X\sim p, Z\sim g_\sigma}[\|v(X+Z) - X\|^2],$$

    with $g_\sigma = \mathcal{N}(0, \sigma^2 I)$. Then

    $$s^\sigma(x) = \nabla \log p^\sigma(x) = \frac{v^\sigma(x) - x}{\sigma^2},$$

    where $p^\sigma(x) = (p \bigotimes g_\sigma)(x) = \int p(y)\Phi_\sigma(x-y)dy$.

  - We can estimate the score $s(x) \approx s^\sigma(x)$ (when σ small) via $\hat{s}_\theta(x) = \frac{\hat{v}_\theta(x) - x}{\sigma^2}$, where the denoising autoencoder

    $$\hat{v}_\theta(x) \in \arg\min_{v_\theta \in \mathcal{F}} \sum_{i=1}^{n} \|v_\theta(X_i + Z_i) - X_i\|^2 / n,$$

    $Z_i, i = 1, ..., n$ are i.i.d. $\mathcal{N}(0, \sigma^2 I)$.

- Bound the error in the framework of analysis of ULA.

Score matching loss

- Stein Lemma
  Let $\xi \sim \mathcal{N}(0, I_d)$ and $f : \mathbb{R}^d \to \mathbb{R}$ be an almost everywhere function with $\mathbb{E}_\xi[\|\nabla f(\xi)\|] < \infty$ and $\lim_{|x_i|\to\infty} f(x)\exp^{-x_i^2/2} = 0$. Then $\mathbb{E}_\xi[f(\xi)\xi] = \mathbb{E}_\xi[\nabla f(\xi)]$.
- Denote $w(x) = v(x) - x$, $Y = X + Z$, some calculation shows
  $\mathbb{E}_X\mathbb{E}_Z[\|v(X+Z) - X\|^2] = \int\int \|V(y) - y + z\|^2 p(y-z)\Phi_\sigma(z)\mathrm{d}y\mathrm{d}z$
  $= \int\int \|w(y)\|^2 p(y-z)\Phi_\sigma(z)\mathrm{d}y\mathrm{d}z + \int \int 2\langle z, w(y)\rangle p(y-z)\Phi_\sigma(z)\mathrm{d}y\mathrm{d}z + C$
- By Stein Lemma, we have

$$\int \langle z, w(z+x)\rangle \Phi_\sigma(z)\mathrm{d}z \quad = \sigma \int \langle \xi, w(x+\sigma\xi)\rangle \Phi_1(\xi)\mathrm{d}\xi$$
$$= \sigma\mathbb{E}_\xi[\langle \xi, w(x+\sigma\xi)\rangle] = \sigma^2\mathbb{E}_\xi[\nabla \cdot w(x+\sigma\xi)] = \sigma^2\mathbb{E}_Z[\nabla \cdot w(x+Z)]$$

- By divergence theorem (assume $\lim_{\|x\|\to\infty} w(x)p^\sigma(x) = 0$)

$$2^{th}\ \text{term} = 2\sigma^2 \int p(x)\mathbb{E}_Z[\nabla \cdot w(x+Z)]\mathrm{d}x = 2\sigma^2\mathbb{E}_{Y\sim p^\sigma}[\nabla \cdot w(Y)]$$
$$= 2\sigma^2 \int \nabla \cdot w(y)p^\sigma(y)\mathrm{d}y = -2\sigma^2 \int \langle w(x), \frac{\nabla p_\sigma(x)}{p_\sigma(x)}\rangle p_\sigma(x)\mathrm{d}x$$
$$= -2\sigma^2\mathbb{E}_{Y\sim p^\sigma}[\langle w(Y), s^\sigma(Y)\rangle]$$

- $$1^{th}\ \text{term} = \int \|w(y)\|^2 (\int p(y-z)\Phi_\sigma(z)\mathrm{d}z)\mathrm{d}y = \mathbb{E}_{Y\sim p^\sigma}[\|w(Y)\|^2].$$
- Score matching loss $\min_{S(\cdot)} \mathbb{E}_{Y\sim p^\sigma}[\|S(Y) - \nabla \log p^\sigma(Y)\|^2]$.

Connection with Schrödinger Bridge

- Algorithm (Inexact Langevin)
  - Input $X_1, ..., X_n$, σ, K.
  - Add noise $Y_i = X_i + \sigma\xi_i$, where $\xi \sim \mathcal{N}(0, I)$.
  - Train

$$\hat{v}_\theta(x) \in \arg\min_{v_\theta \in \mathcal{F}} \sum_{i=1}^{n} \|v_\theta(Y_i) - X_i\|^2/n,$$

  - Sampling new data
    - Do the following iteration K times
      $X_{k+1} = X_k + \gamma_k(\hat{v}_\theta(X_k) - X_k)/\sigma^2 + \sqrt{2\gamma_k}\xi_k, k = 0, 1, ..., K-1.$

- Sampling via ULA with annealing on σ.

- Deeper understanding in the framework of Schrödinger Bridge
  Wang-**Jiao**-Xu-Wang-Yang ***ICML*** 21.

  - SDE

$$\mathrm{d}X_t = \sigma^2 s^{\sqrt{1-t}\sigma}(X_t)\mathrm{d}t + \sigma \mathrm{d}B_t \tag{7}$$

    with $X_0 \sim p^\sigma$ satisfying $X_1 \sim p$.

    - Train a score network $\hat{s}_\theta(t_i, x)$ to estimate $s^{\sqrt{1-t_i}\sigma}(x)$, $t_i = i/K, i = 1, ..., K-1$.
    - Using E-M on the plug in SDE for generate new sample.
    - End-2-end consistency.

Diffusion model via time reversal

- Consider the OU process $\mathrm{d}X_t = -X_t\mathrm{d}t + \sqrt{2}\mathrm{d}B_t, X_0 \sim p$ with explicit solution $X_t = e^{-t}X_0 + e^{-t}\int_0^t \sqrt{2}e^s \mathrm{d}B_s \sim \mathcal{N}(e^{-t}X_0, 1 - e^{-2t})$.

- After time change $\phi(t) = 1 - e^{-t}$, $\overline{X}_t = X_{\phi(t)}$ satisfying

$$\overline{X}_t = (1-t)X_0 + (1-t)\int_0^t \sqrt{2/(1-s)^3}\mathrm{d}B_s,\ t \in [0,1)$$

  is a unique strong solution of $\mathrm{d}X_t = \frac{X_t}{t-1}\mathrm{d}t + \sqrt{\frac{2}{1-t}}\mathrm{d}B_t$.

- For $\epsilon \in (0,1]$, we can reverse $\overline{X}_t$ as $X_t^* = \overline{X}_{1-t}, t \in [\epsilon, 1]$.

- $\overline{X}_t, t \in [0, 1-\epsilon]$ satisfies all conditions of the paper Haussmann-Pardoux 86, so $X_t^* = \overline{X}_{1-t}, t \in [\epsilon, 1]$ satisfying

$$\mathrm{d}X_t^* = [X_t^* + 2\nabla \log p_{1-t}(X_t^*)]/t\mathrm{d}t + \sqrt{2/t}\mathrm{d}B_t,$$

  where $p_t(x) = \int p(y)\Phi_{\sqrt{t(2-t)}}(x - (1-t)y)\mathrm{d}y$. And $X_1^* \sim p$.

  - Loss for estimated the score function $s(t,x) \in \arg\min_s \mathbb{E}_{t\sim \mathrm{U}(0,1), X_t \sim p_t}[\|s(t,X_t) - \nabla \log p_t(X_t)\|^2] \Longleftrightarrow s(t,x) \in \arg\min_s \mathbb{E}_{t\sim \mathrm{U}(0,1), X\sim p, \xi\sim\mathcal{N}(0,I)}[\frac{1}{t(2-t)}\|\sqrt{t(2-t)}s(t,(1-t)X + \sqrt{t(2-t)}\xi) + \xi\|^2] + C$.

  - Train score networks to estimate $s(t,x)$ and use E-M on the plug in SDE to sample Song etal., 21.

One step generation/sampling

Distribution-value interpolation

- Geodesic interpolation Monge 1781, Kantorovich 1942, Benamou-Brenier 2000
  - $$\mathcal{W}_2(\mu,\nu) = \inf_{\mu_t, v_t} \{ \int_0^1 \mathbb{E}_{X\sim\mu_t}[\|v_t(X)\|^2]dt \}^{1/2}$$

    s.t.

    $$\frac{d}{dt}\mu_t = -\nabla\cdot(\mu_t v_t(x)), \quad \mu_0 = \mu, \quad \mu_1 = \nu.$$

- Entropy interpolation Schrödinger 1932, Föllmer 1987
  - Finding a path measure $\mathbf{Q}^* \in \mathcal{P}(\Omega)$ with marginal $\mathbf{Q}_t^*, t \in [0,1]$ such that

    $$\mathbf{Q}^* \in \arg\min \mathbb{D}_{\mathrm{KL}}(\mathbf{Q}||\mathbf{W}_\gamma),$$

    and

    $$\mathbf{Q}_0 = \mu, \mathbf{Q}_1 = \nu.$$

- Stochastic interpolation Albergo-Boffi-Vanden Eijnden 2023
  - Define

    $$\mathbb{X}_t = a(t)\mathbb{X}_0 + b(t)\mathbb{X}_1, \tag{8}$$

    where $\mathbb{X}_0 \sim \mu, \mathbb{X}_1 \sim \nu$, $a(t)$ and $b(t)$ are C^1 functions of $t \in [0,1]$ satisfying

    $$\begin{aligned} &a'(t) \le 0, \quad b'(t) \ge 0, \quad a(0) = 1, \quad a(1) = 0, \quad b(0) = 0, \quad b(1) = 1, \\ &a(t) > 0 \text{ over } t \in [0,1), \quad b(t) > 0 \text{ over } t \in (0,1]. \end{aligned} \tag{9}$$

    $\mu_t = \mathrm{Law}(\mathbb{X}_t), t \in [0,1]$.

Gaussian interpolation flow

- $\nu = \gamma_{d,\sigma} * \rho$, where ρ is a probability measure supported on a ball of radius R.

- Define the ODE by

$$\mathrm{d}X_t = V(t, X_t)\mathrm{d}t \quad t \in [0,1] \tag{10}$$

with $X_0 \sim \mu = \mathrm{Law}(a_0 Z + b_0 \mathbb{X}_1)$, where

$$V(t,x) = \mathbb{E}[\dot{a}_t Z + \dot{b}_t \mathbb{X}_1 | a_t Z + b_t \mathbb{X}_1 = x], t \in (0,1),$$

$$Z \sim \gamma_d, V(0,x) = \lim_{t \downarrow 0} V(t,x), V(1,x) = \lim_{t \uparrow 1} V(t,x).$$

-

Type	VE	VP	Linear	Föllmer	Trigonometric	Square-root
a_t	σ_t	α_t	$1-t$	$\sqrt{1-t^2}$	$\cos(\frac{\pi}{2}t)$	$\sqrt{1-t}$
b_t	1	$\sqrt{1-\alpha_t^2}$	t	t	$\sin(\frac{\pi}{2}t)$	$\sqrt{t}$
a_0	σ_0	α_0	1	1	1	1
b_0	1	$\sqrt{1-\alpha_0^2}$	0	0	0	0
Source	Convolution	Convolution	γ_d	γ_d	γ_d	γ_d

$a_t, b_t \in C^2, t \in (0,1), \dot{a}_t \leq 0, \dot{b}_t \geq 0,$
$a_0 > 0, b_0 \geq 0, a_1 = 0, b_1 = 1, a_t, b_t > 0, t \in (0,1)$

- Generalization of stochastic interpolation Albergo-Vanden-Eijnden 23.

Gaussian interpolation flow

- Theorem Gao-Huang-**Jiao** ***JMLR*** 24

  $X_{t\#}\mu = \mathrm{Law}(a_t Z + b_t \mathbb{X}_1)$, $t \in [0,1]$, especially, $X_{1\#}\mu = \nu$. Furthermore, $X_1(x)$ is Lipschitz with constant $\frac{\sigma}{\sqrt{a_0^2+\sigma^2 b_0^2}} \exp(\frac{a_0^2}{a_0^2+\sigma^2 b_0^2} \frac{R^2}{2\sigma^2})$.

- Applications in applied probability

  - When $a_0 = \sigma_0, b_0 = 1$, we give an explicit Lipschtiz map from $\gamma_{d,\sigma_0} * \nu$ to ν (implicit construction in Klartag-Putterman 2021).

  - ν satisfies logarithmic Sobolev and Poincaré inequality with improved constant Bardet-Gozlan-Malrieu-Zitt 18, Chen-Chewi-Niles-Weed 21.

Sampling with Föllmer flow

- We illustrate with example by setting $\mu = \gamma_d$, $a(t) = \sqrt{1-t^2}$, $b(t) = t$, at this time we call μ_t Föllmer flow.
- In sampling where the normalized constant C in $\nu = \frac{\exp^{-U(x)}}{C}\mathrm{d}x$ is unknown.
- **Idea**: we simulate the ODE (10) via forward Euler by estimating $V(t,x)$ with MC.
- Some calculation shows the vector fields in ODE (10) reads

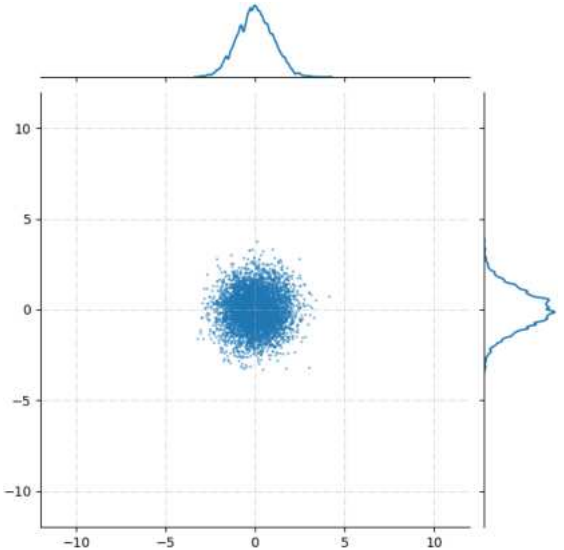
$$V(t,x) = \frac{\mathbb{E}_{Z\sim\gamma_d}[Zg(tx+\sqrt{1-t^2}Z)]}{\mathbb{E}_{Z\sim\gamma_d}[g(tx+\sqrt{1-t^2}Z)]\sqrt{1-t^2}}, \quad g(x) = \exp^{-U(x)+\|x\|^2/2}.$$

- Algorithm
  - Let
$$t_k = k\cdot s, \quad k = 0,1,\ldots,K, \quad \text{with} \quad s = 1/K, \quad X_{t_0} \sim \gamma_d.$$
  - $X_{t_{k+1}} = X_{t_k} + s\widehat{V}_m\left(X_{t_k}, t_k\right)$, $k = 0,1,\ldots,K-1$, where $\widehat{V}_m(x,t)$ is a MC estimator of $V(x,t)$ with m i.i.d Gaussian samples.
- Theorem Ding-**Jiao**-Lu-Yang-Yuan arXiv:2311.03660
  Let $r(x) = \frac{\mathrm{d}\nu}{\mathrm{d}\gamma_d}(x)$. Assume $r, \nabla r$ are Lipschtiz and $r \geq c > 0$. Then,

$$\mathrm{Wass}_2(\mathrm{Law}(X_{t_K}), \nu) \leq \mathcal{O}((d/K^2)^{1/3}) + \mathcal{O}(d/mK).$$

Numerics

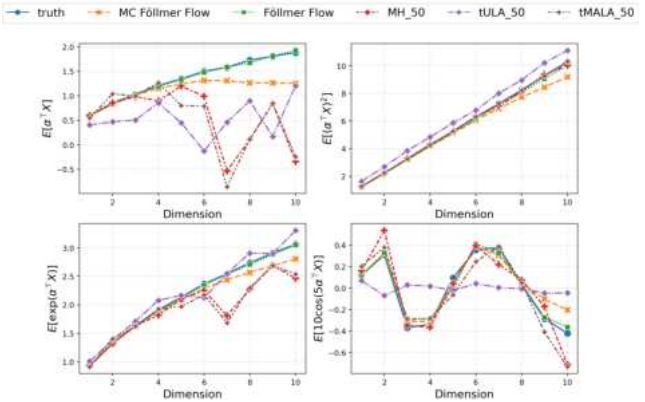
- Particle evolution form Gaussian to mixture of Gaussian



Higher dimensions

- Monte Carlo Föllmer flow with $m = 200d$, $d = 1, \cdots, 10$:

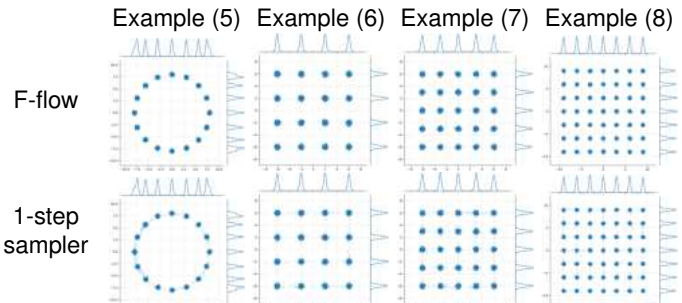
$$\nu(x) = 0.2\mathcal{N}(x; -\alpha_d, 0.25\mathbf{I}_d) + 0.8\mathcal{N}(x; \alpha_d, 0.25\mathbf{I}_d), \ \alpha_d = \mathbf{1}^\top \in \mathbb{R}^d.$$



Monte Carlo estimates of $\mathbb{E}[h(x)]$ versus d for d-dimensional multivariate Gaussian mixture distributions of X, for d increasing from 1 to 10 with lag 1.

One step sampler

- Deterministic transform (inverse transform);
- Training $\hat{g}_\psi \in \arg\min_{g_\psi \in \mathcal{G}} \mathcal{L}(g_\psi) := \sum_{i=1}^{N} \|g_\psi(X_{t_0}^i) - X_{t_K}^i\|_2^2 / N$.



Generative learning with Föllmer flow

- For generative learning, we have $X_1, \ldots, X_n$ i.i.d. form unknown target $\nu \in \mathcal{P}(\mathbb{R}^d)$. We want to learn ν.

- **Idea**: we first learn $V(t,x)$ in ODE (10) and simulate the ODE with an estimated $\widehat{V}(x,t)$ via forward Euler.

  - $V(0,x) = \mathbb{E}_{\mathbb{Y}\sim\nu}[\mathbb{X}]$, then, $\widehat{V}(0,x) = \sum_{i=1}^n X_i/n$.

  - Some calculation shows that $V(t,x), t \in (0,1)$ satisfying

$$\begin{aligned} V(t,x) &= \arg\min_v \mathcal{L}(v(t,x)) \\ &= \mathbb{E}_{t\sim U(0,1), Z\sim\gamma_d, \mathbb{X}_1\sim\nu}[\|(\mathbb{X}_1 - \frac{t}{\sqrt{1-t^2}}Z) - v(t, \sqrt{1-t^2}Z + t\mathbb{X}_1)\|^2]. \end{aligned}$$

  - Estimating $V(t,x)$ with deep neural networks via

$$\widehat{V}_\theta(t,x) = \arg\min_{V_\theta} \widehat{\mathcal{L}}(V_\theta) = \frac{1}{n}\sum_{i=1}^n \|(X_i - \frac{t_i}{\sqrt{1-t_i^2}}Z_i - V_\theta(t_i, \sqrt{1-t_i^2}Z_i + t_i X_i)\|^2, \tag{11}$$

  where $t_i \sim U(0,1), Z_i \sim \gamma_d$.

Generative leaning with Föllmer flow

- Algorithm (Föllmer flow)
  - Let $t_k = k \cdot s, \quad k = 0, 1, \ldots, K, \quad$ with $\quad s = (1 - a)/K, \quad X_{t_0} \sim \gamma_d$.
  - $X_{t_{k+1}} = X_{t_k} + s\widehat{V}_\theta\left(t_k, X_{t_k}\right), k = 0, 1, \ldots, K-1.$
- Algorithm (Characteristic learning for one step generation)
  - Run Above N times in parallel with initial $X_{t_0}^{(i)}$ to get $X_{t_k}^{(i)}, i = 1, ..., N, k = 1, ...K$.
  - Train $\hat{g}_\psi(t, x) \in \arg\min_{g_\psi} \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{\ell=k+1}^{K} \frac{1}{2NK^2} \|g_\psi(t_\ell, X_{t_{k-1}}^{(i)}) - X_{t_\ell}^{(i)}\|^2$.
- Main results

  If we set network structures and N, K, a properly,
  - **Theorem** Chang-Ding-**Jiao**-Li-Yang arXiv:2402.01460, Gao-Huang-**Jiao**-Zheng arXiv:2404.00551

    $$\mathbb{E}[\mathcal{W}_2(\mathrm{Law}(X_{t_K}), \nu(X))] \leq \mathcal{O}(n^{-1/(d+5)}).$$
  - **Theorem** Ding-Duan-**Jiao**-Li-Yang-Zhang arXiv:2405.05512

    $$T \in (0,1), \quad \mathbb{E}[\frac{2}{T^2} \int_0^T \int_t^T \mathcal{W}_2^2(\hat{g}(s,\cdot)_\# \mu_t, \mu_s) \mathrm{d}s\mathrm{d}t] \leq \mathcal{O}(n^{-1/(7d+13)}).$$

Generative learning in latent spaces

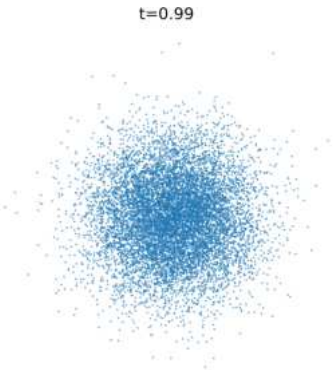
- Extension: latent Föllmer flow
  - Pretrain an auto-encoder $\widehat{E}, \widehat{D}$ on $\tilde{X}_i$ i.i.d $\tilde{\nu}$, $i = 1, ..., m$ via

$$(\widehat{E}, \widehat{D}) \in \arg \min_{E:\mathbb{R}^d \to \mathbb{R}^{d^*},\ D:\mathbb{R}^{d^*} \to \mathbb{R}^d} \widehat{\mathcal{L}}_{\mathrm{AE}} = \sum_{i=1}^{m} \frac{1}{m} \|D(E(\tilde{X}_i)) - \tilde{X}_i\|_2^2. \tag{12}$$

  - Get latent data $\bar{X}_i = \widehat{E}(X_i), i = 1, ...n$.
  - Estimate $\widehat{V}_\theta$ on $\bar{X}_i, i = 1, ...n$.
  - Run Algorithm Föllmer flow with initial $\bar{X}_{t_0} \sim \gamma_{d^*}$ to get $\bar{X}_{t_k} k = 1, ...K$.
  - Generation in data space $X_{t_k} = \widehat{D}(\bar{X}_{t_k}), k = 1, ...K$.
  - **Theorem** **Jiao**-Lai-Wang-Yan arXiv:2404.02538
    Define $(E^*, D^*) \in \arg\min_{E:\mathbb{R}^d \to \mathbb{R}^{d^*},\ D:\mathbb{R}^{d^*} \to \mathbb{R}^d} \mathcal{L}_{\mathrm{AE}}$. Under certain regularity condition and proper set the hyperparameters,

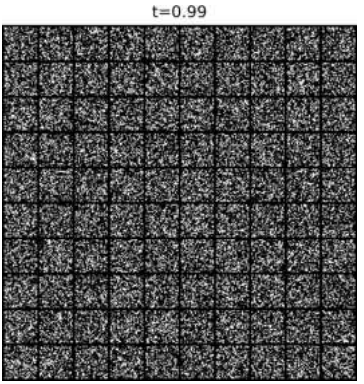
$$\begin{aligned}\mathbb{E}[\mathcal{W}_2(\nu, \mathrm{Law}(X_{t_K}))] \leq\ & \mathcal{O}(n^{-1/4(d^*+5)}) + \mathcal{O}(m^{-1/(d+2)}) \\ & + \mathcal{O}(\mathrm{Lip}(\widehat{D})\mathrm{Lip}(\widehat{E})\mathcal{W}_2(\nu, \tilde{\nu})) + \mathcal{O}(\mathcal{L}_{\mathrm{AE}}(E^*, D^*)).\end{aligned}$$

Numerics: swissroll via characteristics learner $\hat{g}_\psi$



$1 - t$ in time

Numerics: MNIST via characteristics learner $\hat{g}_\psi$



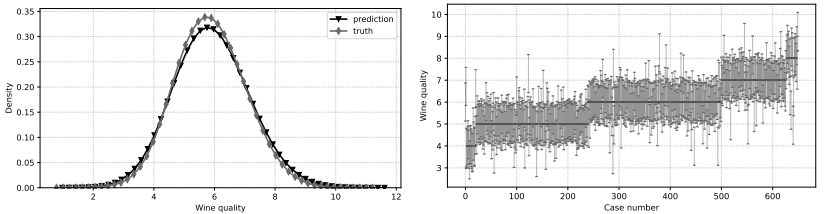
$1 - t$ in time

Performance comparisons on CIFAR-10

Model	NFE ↓	FID ↓
GAN Models		
BigGAN	1	8.51
StyleGAN-Ada	1	2.92
Diffusion + Sampler		
DDPM	1000	3.17
DDIM	100	4.16
Diffusion + Distillation		
Rectified Flow	1	4.85
PD	1	9.12
CD	1	10.53
CTM (without GAN)	1	5.19
CG (ours)	1	**4.59**
PD	2	4.51
CTM (without GAN)	18	3.00
CG (ours)	2	**3.50**
CG (ours)	4	**2.83**

Model free prediction with UQ

- Uncertainty quantification via learn conditional distribution $\nu(\cdot|X = x)$.
  - Conditional GAN Zhou-**Jiao**-Liu-Huang ***JASA*** 23, conditional diffusion model **Jiao**-Kang-Liu-Peng-Zuo ***TIT*** 25+, conditional ODE flow Chang-Ding-**Jiao**-Li-Yang arXiv:2402.01460 with paired data $(\mathbf{X}_i, \mathbf{Y}_i)_{i=1,\ldots n}$.
- The wine quality dataset (UCI machine learning repository) $(X_i, Y_i), i = 1..., 6497$. with $X_i \in \mathbb{R}^{11}$. We use 75% of the data for training, 15% for validation, and 10% for testing.



(left) Estimated conditional density and actual density of the test set. (right) The prediction intervals on the test set.

- We construct the 90% prediction interval on the test set shown in the right of the figure. The actual coverage for all 650 cases in the test set is 91.23%.

Outline

Conclusion and discussion

- Error analysis of DRM with overparametrization
  - New error decomposition
    - Control $\mathcal{E}_{\text{opt}}^{-}$ under overparametrization without neither NTK nor mean fields theory (not lazy training scheme !).
- Simultaneously function and derivative estimation via deep Sobolev regression.
- Schemes from GAN to diffusion models.
- Error of GAN.
- Error of latent ODE flow.
- Provable one-step generative learning.

THANK YOU FOR YOUR ATTENTION!

Yuling Jiao

http://jszy.whu.edu.cn/jiaoyuling/en/index.htm