Deep Generative Learning via Euler Particle Transport

Yuan Gao [*] Jian Huang [†] Yuling Jiao [‡] Jin Liu [§] Xiliang Lu [‡] Zhijian Yang [‡]

[*] Xi'an Jiaotong University

[†] University of Iowa

[‡] Wuhan University

[§] Duke-NUS Medical School

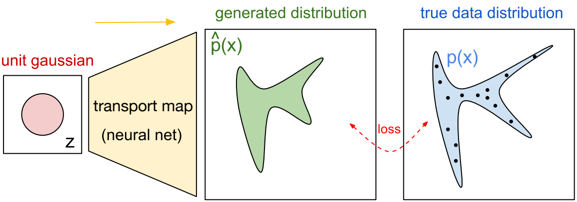
July 31, 2021

Generative learning

Generative learning: learn representations of probability distributions from target data

- directly represent the sampling process
- represent a probability density function

Deep generative learning: generative learning with deep neural networks
Motivation: learn a transport map to represent the sampling process from target data
Solution: optimal transport and gradient flows



Deep generative learning via the transport map.

Optimal transport

Let μ and ν be two probability measures. Suppose $Z \sim \mu$. Denote the distribution of $T(Z)$ by $T_{\#}\mu$, the pushforward of the measure μ under T. Then T is called a transport from μ to ν if

$$T_{\#}\mu = \nu.$$

Monge problem

Find a transport T of the probability mass under μ to ν minimizing the quadratic cost,

$$\min_{\mathcal{T}:\mathcal{T}_{\#}\mu=\nu} \frac{1}{2}\mathbb{E}_{X\sim\mu}\|X - \mathcal{T}(X)\|^2. \tag{1}$$

Any map $\mathcal{T}$ that is a solution of (1) is called an optimal transport map.

Kantorovich problem

To resolve the existence issue of the Monge problem (1), Kantorovich introduced a relaxation of (1),

$$\mathcal{W}_2(\mu, \nu) = \{\inf_{\gamma\in\Gamma(\mu,\nu)} \mathbb{E}_{(X,Y)\sim\gamma}[\|X - Y\|_2^2]\}^{\frac{1}{2}}, \tag{2}$$

where $\Gamma(\mu, \nu)$ denotes the set of couplings of (μ, ν) [7, 2].

Optimal transport

Suppose that μ and ν have densities q and p with respect to the Lebesque measure, respectively.

- The minimization problem in (2) admits a unique solution $\gamma = (\boldsymbol{I}, \mathcal{T})_{\#}\mu$ with $\mathcal{T} = \nabla\Psi$, where $\boldsymbol{I}$ is the identity map and $\nabla\Psi$ satisfies the Monge-Ampère equation

$$\det(\nabla^2\Psi(\boldsymbol{x})) = \frac{q(\boldsymbol{x})}{p(\nabla\Psi(\boldsymbol{x}))}, \boldsymbol{x} \in \mathbb{R}^m. \tag{3}$$

- To find the optimal transport $\mathcal{T}$, it suffices to solve (3) for Ψ.
- However, this equation is difficult to solve due to the high nonlinearity of $\det$.

Optimal transport: linearization

- Due to the high nonlinearity of det, we consider the linearized form of the Monge-Ampère equation [7]

$$\Psi(\boldsymbol{x}) = \|\boldsymbol{x}\|^2/2 + t\Phi(\boldsymbol{x}), t \geq 0,$$

thus

$$\nabla\Psi(\boldsymbol{x}) = \boldsymbol{x} + t\nabla\Phi(\boldsymbol{x}).$$

- Let $t \to 0$, we get the random process $\{\mathbf{X}_t\}$ and its laws $\{q_t\}$ satisfying

$$\begin{aligned} \frac{d\mathbf{X}_t}{dt} &= \nabla\Phi(\mathbf{X}_t), \ t \geq 0 \\ \frac{d \ln q_t(\boldsymbol{x})}{dt} &= -\triangle\Phi(\boldsymbol{x}) \end{aligned}$$

with

$$\mathbf{X}(0) = Z, q_0 = \mu = p_Z, \text{ and } q_\infty = \gamma = p_X,$$

where $\triangle$ is the Laplacian operator:

$$\triangle f = \sum_{i=1}^{m} \frac{\partial^2 f}{\partial x_i^2}.$$

Linearization and McKean-Vlasov equation

A basic approach to addressing the difficulty due to nonlinearity is linearization.

- We use a linearization method based on the residual map

$$\mathcal{T}_{t,\Phi_t} = \nabla \Psi = \mathbb{1} + t\nabla\Phi_t, t \geq 0, \tag{4}$$

  where $\Phi_t : \mathbb{R}^m \to \mathbb{R}^1$ is a function to be chosen such that the law of $\mathcal{T}_{t,\Phi_t}(Z)$ approaches ν as t increases [7].

- This linearization scheme leads to the stochastic process $\mathbf{X}_t$ satisfying the McKean-Vlasov equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{X}_t(\boldsymbol{x}) = \boldsymbol{v}_t(\mathbf{X}_t(\boldsymbol{x})),\ t \geq 0,\ \text{with}\ \ \mathbf{X}_0 \sim \mu,\ \mu\text{- a.e. } \boldsymbol{x} \in \mathbb{R}^m, \tag{5}$$

  where $\boldsymbol{v}_t$ is the velocity vector field of $\mathbf{X}_t$.

- We have $\boldsymbol{v}_t = \nabla\Phi_t$. Thus $\boldsymbol{v}_t$ also determines the residual map (4).

Gradient flow

- The movement of the particles $\{\mathbf{X}_t\}_{t\geq 0}$ along t is completely governed by the velocity fields $\boldsymbol{v}_t$, given the initial value.
- We choose a $\boldsymbol{v}_t$ to decrease the discrepancy, e.g., f-divergence, between the distribution of $\mathbf{X}_t$, say μ_t, at time t and the target ν.
- An equivalent formulation of (5) is through the gradient flow $\{\mu_t\}_{t\geq 0}$, where $\mathbf{X}_t \sim \mu_t$, with $\{\boldsymbol{v}_t\}_{t\geq 0}$ as its velocity fields:

$$\frac{\partial}{\partial t}\mu_t = -\nabla \cdot (\mu_t \boldsymbol{v}_t) \text{ in } \mathbb{R}^+ \times \mathbb{R}^m \text{ with } \mu_0 = \mu,$$

- Computationally it is more convenient to work with the Mckean-Vlasov equation (5).

Velocity fields

- The basic intuition is that we want to move along the direction that reduces the discrepancies between μ_t and the target ν.
- We use *f*-divergence [1] to measure the discrepancies:

$$\mathcal{L}[\mu_t] = \mathbb{D}_f(\mu_t \| \nu) = \int_{\mathbb{R}^m} p(\boldsymbol{x}) f\left(\frac{q_t(\boldsymbol{x})}{p(\boldsymbol{x})}\right) \mathrm{d}\boldsymbol{x},$$

  where q_t is the density of μ_t, p is the density of ν and $f : \mathbb{R}^+ \to \mathbb{R}$ is a convex function with $f(1) = 0$.
- We choose the velocity fields $\boldsymbol{v}_t$ such that $\mathcal{L}[\mu_t]$ is minimized. This leads to

$$\boldsymbol{v}_t(\boldsymbol{x}) = \Phi_t(\boldsymbol{x}) = -\nabla f'(r_t(\boldsymbol{x})), \quad \text{where} \quad r_t(\boldsymbol{x}) = \frac{q_t(\boldsymbol{x})}{p(\boldsymbol{x})}, \quad \boldsymbol{x} \in \mathbb{R}^m.$$

- If we use the χ^2-divergence with $f(c) = (c-1)^2/2$, then

$$\boldsymbol{v}_t(\boldsymbol{x}) = \nabla r_t(\boldsymbol{x})$$

  is simply the gradient of the **density ratio**.

Euler particle transport

- We discretize the McKean-Vlasov equation (5). Let $s > 0$ be a small step size. We use the forward Euler method defined iteratively by:

$$\mathcal{T}_k = \mathbb{1} + s\boldsymbol{v}_k, \tag{6}$$

$$\mathbf{X}_{k+1} = \mathcal{T}_k(\mathbf{X}_k) = \mathbf{X}_k + s\boldsymbol{v}_k(\mathbf{X}_k), \tag{7}$$

  where $\mathbf{X}_0 \sim \mu$, $\mu_0 = \mu$, $\boldsymbol{v}_k$ is the velocity field at the kth step, $k = 0, 1, ..., K$ for some large K.

- The final transport map is

$$\mathcal{T} = \mathcal{T}_K \circ \mathcal{T}_{K-1} \cdots \circ \mathcal{T}_0,$$

  which is the composition of a sequence of simple residual maps $\mathcal{T}_K, \ldots, \mathcal{T}_1, \mathcal{T}_0$.

- We refer to this updating scheme as the **Euler particle transport (EPT)**.

Training Euler transport map

- When a random sample is available, it is natural to learn ν by first estimating the velocity fields $\boldsymbol{v}_k$ and then plugging the estimated $\boldsymbol{v}_k$ in (6).
- If we use the f-divergence as the energy functional, estimating the velocity fields

$$\boldsymbol{v}_k(\boldsymbol{x}) = -\nabla f'(r_k(\boldsymbol{x})),$$

  boils down to estimating the density ratios

$$r_k(\boldsymbol{x}) = \frac{q_k(\boldsymbol{x})}{p(\boldsymbol{x})}$$

  dynamically at each iteration $k = 1, \ldots, K$.
- We **estimate density ratios nonparametrically** using Bregman divergences and gradient regularizer
- Let $\hat{\boldsymbol{v}}_k$ be the estimated velocity fields at the kth iteration. The kth estimated residual map is $\widehat{\mathcal{T}}_k = \mathbb{1} + s\hat{\boldsymbol{v}}_k$. Finally, the trained map is

$$\widehat{\mathcal{T}} = \widehat{\mathcal{T}}_K \circ \widehat{\mathcal{T}}_{K-1} \circ \cdots \circ \widehat{\mathcal{T}}_0.$$

Error bounds

- Error due to linearization of the Monge-Ampère equation

$$\mathcal{W}_2(\mu_t, \nu) = \mathcal{O}(e^{-\lambda t}),$$

  for some $\lambda > 0$. Therefore, μ_t converges to ν exponentially fast as $t \to \infty$.
- Discretization: For an integer $K \geq 1$ and a small $s > 0$, let

$$\{\mu_t^s : t \in [ks, (k+1)s), k = 0, \ldots, K\}$$

  be a piecewise constant interpolation between μ_{ks} and $\mu_{(k+1)s}, k = 0, 1, \ldots, K$.
- Error due to discretization of μ_t^s in a finite time interval $[0, T]$ can be bounded :

$$\sup_{t \in [0,T]} \mathcal{W}_2(\mu_t, \mu_t^s) = \mathcal{O}(s).$$

Density-ratio estimation

- Let $r(\boldsymbol{x}) = q(\boldsymbol{x})/p(\boldsymbol{x})$ be the density ratio.
- Let $g : \mathbb{R} \to \mathbb{R}$ be a differentiable and strictly convex function.

Bregman score

The Bregman score with the base probability density p for measuring the discrepancy between r and a measurable function $R : \mathbb{R}^m \to \mathbb{R}^1$ is

$$\mathfrak{B}(r, R) = \mathbb{E}_{X\sim p}[g'(R(X))R(X) - g(R(X))] - \mathbb{E}_{X\sim q}[g'(R(X))].$$

Least-squares density-ratio fitting

The least squares density-ratio (LSDR) fitting criterion with $g(c) = (c-1)^2$ is

$$\mathfrak{B}_{\mathrm{LSDR}}(r, R) = \mathbb{E}_{X\sim p}[R(X)^2] - 2\mathbb{E}_{X\sim q}[R(X)] + 1.$$

Density-ratio estimation

LSDR estimation with gradient regularizer

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

- Let $\mathcal{H} \equiv \mathcal{H}_{\mathcal{D},\mathcal{W},\mathcal{S},\mathcal{B}}$ be the set of ReLU neural networks R_ϕ with parameter ϕ, depth $\mathcal{D}$, width $\mathcal{W}$, size $\mathcal{S}$, and $\|R_\phi\|_\infty \leq \mathcal{B}$.
- We combine the LSDR loss with the gradient regularizer as our objective function.

LSDR estimator

The resulting gradient regularized LSDR estimator of $r = p/q$ is given by

$$\widehat{R}_\phi \in \arg\min_{R_\phi \in \mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}[R_\phi(X_i)^2 - 2R_\phi(Y_i)] + \alpha\frac{1}{n}\sum_{i=1}^{n}\|\nabla R_\phi(X_i)\|_2^2, \tag{8}$$

where $\alpha \geq 0$ is a regularization parameter.

Density-ratio estimation

Next we bound the nonparametric estimation error of the density-ratio estimator under the assumptions that the support of $\nu \equiv P_X$ is concentrated on a compact low-dimensional manifold and r is Lipsichiz continuous.

- Let $\mathfrak{M} \subseteq [-c, c]^m$ be a Riemannian manifold [4] with dimension $\mathfrak{m}$, condition number $1/\tau$, volume $\mathcal{V}$, geodesic covering regularity $\mathcal{R}$, and $\mathfrak{m} \ll \mathcal{M} = \mathcal{O}\left(\mathfrak{m} \ln(m\mathcal{V}\mathcal{R}/\tau)\right) \ll m$.
- Denote $\mathfrak{M}_\epsilon = \{\boldsymbol{x} \in [-c, c]^m : \inf\{\|\boldsymbol{x} - \boldsymbol{y}\|_2 : \boldsymbol{y} \in \mathfrak{M}\} \leq \epsilon\}$, $\epsilon \in (0, 1)$.

Theorem 1

- *Assume* $\mathrm{supp}(r) = \mathfrak{M}_\epsilon$ *and* $r(\boldsymbol{x})$ *is Lipschitz continuous with the bound* B *and the Lipschitz constant* L.
- *Suppose the topological parameter of* $\mathcal{H}_{\mathcal{D},\mathcal{W},\mathcal{S},\mathcal{B}}$ *in (8) with* $\alpha = 0$ *satisfies* $\mathcal{D} = \mathcal{O}(\log n)$, $\mathcal{W} = \mathcal{O}(n^{\frac{\mathcal{M}}{2(2+\mathcal{M})}} / \log n)$, $\mathcal{S} = \mathcal{O}(n^{\frac{\mathcal{M}-2}{\mathcal{M}+2}} / \log^4 n)$, *and* $\mathcal{B} = 2B$.

Then,

$$\mathbb{E}_{\{X_i, Y_i\}_{i=1}^n}[\|\widehat{R}_\phi - r\|_{L^2(\nu)}^2] \leq C(B^2 + cLm\mathcal{M})n^{-2/(2+\mathcal{M})},$$

where C *is a universal constant.*

Density-ratio estimation

This result is of independent interest for nonparametric estimation with deep neural networks. The error bound established in Theorem 1 for the nonparametric deep density-ratio fitting is new.

- If the intrinsic dimension $\mathcal{M}$ of the data is much smaller than the ambient dimension m, the convergence rate
$$\mathcal{O}(n^{-\frac{2}{2+\mathcal{M}\log d}})$$
is faster than the optimal rate of convergence for nonparametric estimation of a Lipschitz target in $\mathbb{R}^d$, where the optimal rate is
$$\mathcal{O}(n^{-\frac{2}{2+d}}),$$
see e.g., [6, 5].
- The proposed density-ratio estimators are capable of circumventing the "curse of dimensionality" if data is supported on a lower-dimensional manifold.
- Low-dimensional latent structure of many complex data has been frequently encountered by practitioners in image analysis, computer vision and natural language processing.

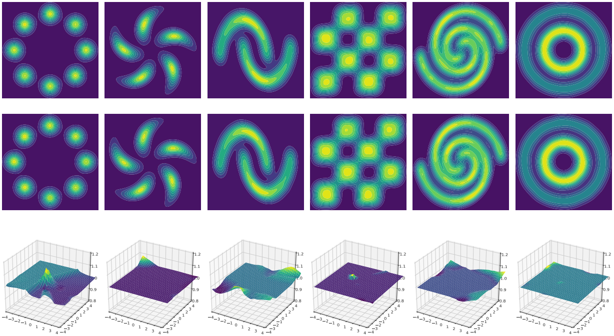
Euler particle transport: algorithm

- **Outer loop for modeling low dimensional latent structure (optional)**
  - Sample $\{Z_i\}_{i=1}^n \subset \mathbb{R}^\ell$ from a low-dimensional reference distribution $\tilde{\mu}$
  - Compute $\tilde{Y}_i = G_\theta(Z_i), i = 1, 2, \ldots, n$.
  - **Inner loop for finding the push-forward map**
    - If there are no outer loops, sample $\tilde{Y}_i \sim \mu, i = 1, \ldots, n$.
    - Get $\hat{\boldsymbol{v}}(\boldsymbol{x}) = -\nabla f'(\widehat{R}_\phi(\boldsymbol{x}))$ via solving (8) with $Y_i = \tilde{Y}_i$. Set $\widehat{\mathcal{T}} = \mathbb{1} + s\hat{\boldsymbol{v}}$ with a small step size s.
    - Update the particles $\tilde{Y}_i = \widehat{\mathcal{T}}(\tilde{Y}_i)$, $i = 1, \ldots, n$.
  - **End inner loop**
  - If there are outer loops, update the parameter θ of $G_\theta(\cdot)$ via solving $\min_\theta \sum_{i=1}^n \|G_\theta(Z_i) - \tilde{Y}_i\|_2^2/n$.
- **End outer loop**

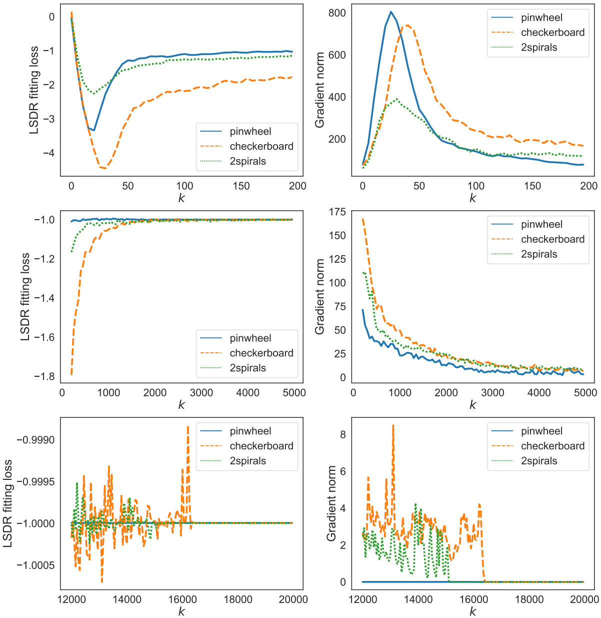
Numerical experiments

- 2-D simulated data.
- Benchmark real data set:
  - MNIST, 60K (28×28)
  - Fashion-MNIST, 60K (28×28)
  - CIFA-10, 50K (32×32)
  - CelebA: 200K (64×64)
- Network architecture/ hyperparameters: see paper.
- Platform: Pytorch with NVIDIA Tesla K80 GPUs.
- The PyTorch code of EPT is available at https://github.com/xjtuygao/EPT.

Numerical experiments: 2-D distributions

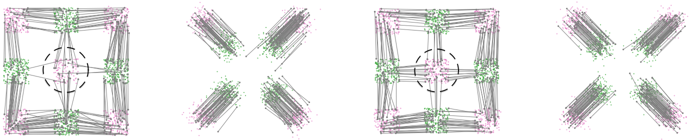


KDE plots of the target samples (the first row) and the corresponding generated samples (the second row). The third row shows surface plots of estimated density ratio after 20k iterations.

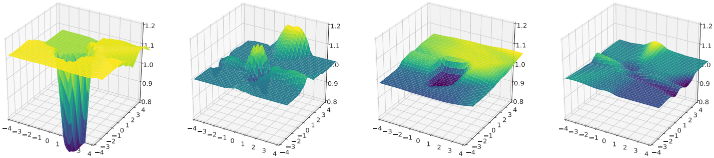


Convergence of EPT on *pinwheel, checkerboard* and *2spirals*. **Top:** The initialization stage. **Middle:** The decline stage. **Bottom:** The converging stage. **Left:** LSDR fitting loss (20) with $\alpha = 0$. **Right:** Estimation of the gradient norm $\mathbb{E}_{X \sim q_k}[\|\nabla R_\phi(X)\|_2]$.

Numerical experiments: 2-D distributions

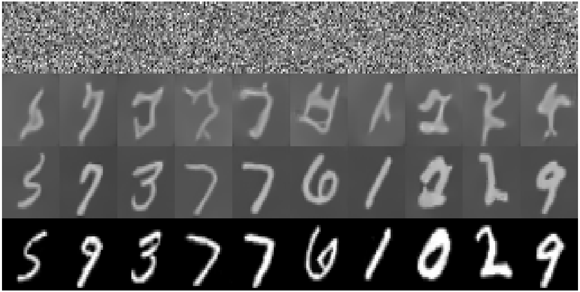


Learned 5*squares* from 4*squares*, and *large*4*gaussians* from *small*4*gaussians*.
Left two figures: Maps learned without gradient penalty. **Right** two figures: Maps learned with gradient penalty.



Left two figures: Surface plots of estimated density-ratio without gradient penalty. **Right** two figures: Surface plots of estimated density-ratio with gradient penalty.

Particle evolution of EPT on MNIST

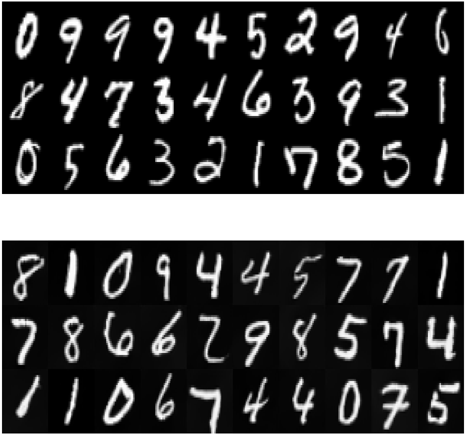


Particle evolution of EPT on MNIST

Particle evolution of EPT on CIFAR10

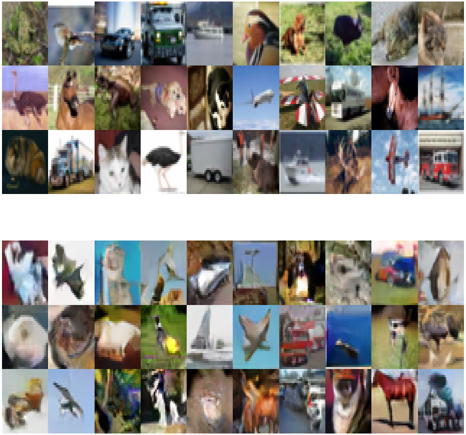
Particle evolution of EPT on CIFAR10.

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

Numerical experiments: FID scores

Mean (standard deviation) of FID scores on CIFAR10 and results in last six rows are adapted from [3].

Models	CIFAR10 (50k)
EPT-LSDR-χ^2	**24.9 (0.1)**
EPT-LR-KL	25.9 (0.1)
EPT-LR-JS	25.3 (0.1)
EPT-LR-logD	**24.6 (0.1)**
WGAN-GP	31.1 (0.2)
MMDGAN-GP-L2	31.4 (0.3)
SMMDGAN	31.5 (0.4)
SN-GAN	26.7 (0.2)
SN-SWGAN	28.5 (0.2)
SN-SMMDGAN	**25.0 (0.3)**

Conclusion

- Generative learning is an effective approach to learning distributions of complex high-dimensional data.
- The key factor for the success of generative learning is the use of deep neural networks to approximate high-dimensional functions nonparametrically.
- The proposed Euler particle transport (EPT) method combines the strength of optimal transport, stochastic differential equation and deep density-ratio estimation.
- EPT is computationally stable and relatively easy to train.
- The numerical performance of ETP is comparable with the state-of-the-art methods.

THANK YOU FOR YOUR ATTENTION!

[1] Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1):131–142, 1966.

[2] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

[3] Michael Arbel, Dougal Sutherland, Mikolaj Binkowski, and Arthur Gretton. On gradient regularizers for MMD GANs. In *NIPS*, 2018.

[4] John Lee. *Introduction to Riemannian Manifolds*. Springer, 2010.

[5] Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *The Annals of Statistics, in press*, 2020.

[6] Charles J. Stone. Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics*, 10(4):1040–1053, 1982.

[7] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.