Touring sampling with pushforward maps

Vivien Cabannes, Charles Arnal

TOURING SAMPLING WITH PUSHFORWARD MAPS

Vivien Cabannes
Meta AI, FAIR Labs
New York, USA

Charles Arnal
INRIA, Datashape team
Sophia-Antipolis, France

ABSTRACT

The number of sampling methods could be daunting for a practitioner looking to cast powerful machine learning methods to their specific problem. This paper takes a theoretical stance to review and organize many sampling approaches in the "generative modeling" setting, where one wants to generate new data that are similar to some training examples. By revealing links between existing methods, it might prove useful to overcome some of the current challenges in sampling with diffusion models, such as long inference time due to diffusion simulation, or the lack of diversity in generated samples.

***Index Terms*—** Ancestral sampling, Pushforward sampler, Statistics Matching, Likelihood maximization, Diffusion models.

1. INTRODUCTION

Generative AI has received much attention recently. It consists in generating realistic data, could it be drugs to cure disease [1], award-winning images [2], or chat bots [3]. McKinsey recently estimated that it could boost the global economy by up to $4.4 trillion annually [4]. Recent advances in generative AI are due to machine learning, where, rather than complex engineering rules, a machine learns to generate new data that resemble training samples. Several generic[1] perspectives have been suggested to do so, ranging from variational auto-encoder, generative adversarial networks (GANs), stochastic diffusion models, or flow models. On the one hand, a GAN provides fast inference but is hard to train; on the other hand diffusion models are easy to train but are slow to generate new samples. Could we have the best of both worlds?

The crux of this paper is to tour the sampling literature to explore and organize principled ways to learn deterministic mappings between distributions, which lead to fast inference algorithms. We distinguish three different perspectives, one based on test functions, one based on density likeliness, and one based on probability flows.

Setting & Notations. This paper considers $\mathcal{X} \subset \mathbb{R}^d$ endowed with a target distribution $\mu_1 \in \Delta_{\mathcal{X}}$, that generates samples $X \sim \mu_1$. Here, Δ_A denotes the set of probability measures over a space A. The goal is to parameterize μ_1 from a user-specified initial distribution $\nu \in \Delta_{\mathcal{Z}}$ for some "sampling" space $\mathcal{Z}$, with a deterministic pushforward map $\varphi : \mathcal{Z} \to \mathcal{X}$. In other terms, we want to learn φ such that one can cast a sample $Z \sim \nu$, which is easy to generate, as a sample $X = \varphi(Z) \sim \mu_1$ following the desired distribution μ_1. Formally, we want $\nu(\varphi^{-1}(A)) = \mu(A)$ for any measurable set $A \subset \mathcal{X}$, which reads $\varphi_{\#}\nu = \mu_1$ when written with pushforward notation. In order to benefit from the extensive literature on transport maps between measures on $\mathcal{X}$, we will eventually lift ν into $\mu_0 \in \Delta_{\mathcal{X}}$ through some canonical embedding $\xi : \mathcal{Z} \to \mathcal{X}$ such that $\xi_{\#}\nu = \mu_0$. The learning of φ is reparameterized as the learning of $\psi : \mathcal{X} \to \mathcal{X}$ that maps μ_0 to μ_1, i.e. $\psi_{\#}\mu_0 = \mu_1$, with $\varphi = \psi \circ \xi$.

[1]We let aside specific perspectives such as next token prediction for LLMs.

2. STATISTICS MATCHING OBJECTIVES

The first approach we will consider leverages the duality bracket, defined for any measurable function $f : \mathcal{X} \to \mathbb{R}$ and measure $\mu \in \mathcal{M}_{\mathcal{X}}$ over $\mathcal{X}$ as

$$\langle f \,|\, \mu \rangle := \int_{\mathcal{X}} f(x)\mu(\mathrm{d}x) =: \mathbb{E}_{X\sim\mu}[f(X)]. \tag{1}$$

When μ is parameterized as $\mu = \varphi_{\#}\nu$, this corresponds to

$$\langle f \,|\, \varphi_{\#}\nu \rangle = \mathbb{E}_{Z\sim\nu}[f(\varphi(Z))].$$

To make sure that $\mu = \mu_1$, one can make sure that the actions of both measures are the same against any test functions (or equivalently that the action of the signed measure $\mu - \mu_1$ is null everywhere).

Maximum formulation. Introducing $\mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ a space of test functions, one can look for the variational objective defined by the worse function

$$D_\infty(\mu, \mu_1; \mathcal{F}) := \sup_{f\in\mathcal{F}} \langle f \,|\, \mu - \mu_1 \rangle. \tag{2}$$

Those measures are known as integral probability metrics [5].

When $\mathcal{F}$ is the unit ball $\mathcal{B}_k$ of a reproducing kernel Hilbert space with kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, (2) becomes the maximum mean discrepancy, or MMD [6]

$$D^2_\infty(\mu, \mu_1; \mathcal{B}_k) = \mathbb{E}[k(X, X') + k(Y, Y') - 2k(X, Y)],$$

where X, X' are independent variables distributed according to μ and Y, Y' follows μ_1. When $\mathcal{F}$ is taken as all the functions from $\mathcal{X}$ to $[0,1]$, it becomes the total variation defined as

$$D_\infty(\mu, \mu_1; [0,1]^{\mathcal{X}}) = \max_{A \subset \mathcal{X}} |\mu(A) - \mu_1(A)|.$$

When f, the argument of the maximum, is learned with an adversarial neural network, this approach identifies to GANs. And when $\mathcal{F}$ is taken as $\mathcal{C}_1^{0,1}$ the set of all contractions, i.e., functions that are 1-Lipschitz continuous, (2) becomes the W^1-Wasserstein metric, which has motivated W-GANs [7].

Guarantees. From those observations, one can deduce that when the span of $\mathcal{F}$ is dense in $\mathcal{C}_1^{0,1}$ (with the W_1^*-topology), D_∞ is a distance. As a consequence, the function $\mu \mapsto D_\infty(\mu, \mu_1)$ would have a unique local minimum, achieved for $\mu = \mu_1$, reachable through gradient descent. In practice, the descent could take place with respect to the parameter θ which parameterizes the map φ_θ, itself parameterizing $\mu = (\varphi_\theta)_\# \nu$. Eventually, the space $\mathcal{F}$ may be chosen to ensure the consistency of the loss $\mu \mapsto D_\infty(\mu, \mu_1)$ for the sole target μ_1.[2]

Mean formulation. When the maximization in (2) does not have a closed-form solution, fitting φ implies solving a min-max optimization problem. This is potentially hard, arguably explaining the instability observed when training GANs [9].[3] Interestingly, one could equally consider some average with respect to a measure τ on $\mathcal{F}$,

$$D_2^2(\mu, \mu_1; \tau) := \int_{\mathcal{F}} \langle f \,|\, \mu - \mu_1 \rangle^2 \tau(\mathrm{d}f). \tag{3}$$

This integrand can be rewritten as $\langle f \,|\, \mu \rangle^2 - 2\langle f \,|\, \mu \rangle \langle f \,|\, \mu_1 \rangle + \langle f \,|\, \mu_1 \rangle^2$, which allows us to approximate the objective $D_2^2(\mu, \mu_1; \tau)$ empirically without bias thanks to the formula

$$\langle f \,|\, \mu \rangle^2 = \mathbb{E}_{X_1, X_2 \sim \mu}[f(X_1) f(X_2)].$$

Once again, when the span of the support of τ is dense in $\mathcal{C}_1^{0,1}$, this defines a metric on measures.

Characteristic function matching. From a theory perspective, it is natural to consider $\mathcal{F} = \{f_t : x \mapsto e^{tx} \,|\, t \in T\}$ for T a subset of $\mathbb{C}$. Those spaces of functions relate to the moment-generating function $t \mapsto \mathbb{E}[e^{tX}] = \sum_{k \in \mathbb{N}} t^k \mathbb{E}[X^k]/k!$ and the characteristic function $t \mapsto \mathbb{E}[e^{itX}]$. Using analytical properties, it turns out that as soon as $T \in \mathbb{C}^{\mathbb{N}}$ has a limit point, $\mathcal{F} = \{f_t \,|\, t \in T\}$ can be used to define a metric through (2) or (3). It does not seems that those spaces have been utilized for generative AI, although we note that the specific case of D_2 where $\mathcal{F} = \mathbb{R} \cdot i$ and $\tau(\mathrm{d}f_{i \cdot s}) = |s|^{-(d+1)}\,\mathrm{d}s$ is known in the literature as the "energy distance",[4] and can be rewritten as a MMD distance with $k(x, y) = \|x - y\|$ [10].

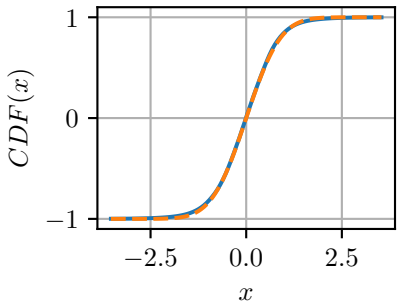

Fig. 1: Learning to map the Gaussian distribution to the uniform one with the mean formulation D_2^2, $\mathcal{F}$ taken as the set of function $\{x \in [0,1] \mapsto e^{2\pi i k x} | k \in [10]\}$, with τ uniform over those ten functions, and φ parameterized with a small multi-layer perceptron. A solution to this problem is given by the cumulative distribution function of the Gaussian, which relates to the error function. The ground truth is plotted in blue, while the learned pushforward is plotted in dashed orange.

Generalization. While we have compared the same statistics on μ and μ_1, one can generalize D_∞ to $\mathcal{J} \subset L^1(\mu) \times L^1(\mu_1)$ under the form

$$D_\infty(\mu, \mu_1; \mathcal{J}) := \sup_{(f,g) \in \mathcal{J}} \langle f \,|\, \mu \rangle - \langle g \,|\, \mu_1 \rangle.$$

Taking the max over $(g, h) \in \mathcal{J}$ allows to cast φ-divergences as D_∞ for $\mathcal{J} = \{(f, \varphi^*(g) | f : \mathcal{X} \to \mathbb{R}\}$ where φ^* is the convex conjugate of f, as per Theorem 7.24 in [11], see also [12, 13]. Moreover, this generalization describes Wasserstein distances thanks to Kantorovich duality,

$$W_c(\mu, \mu_1) := \min_{\pi \in \Pi_{\mu,\mu_1}} \mathbb{E}_{(x,y) \sim \pi}[c(x, y)] = D_\infty(\mu, \mu_1; \mathcal{J}_\Pi),$$

where $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ is a cost function, Π_{μ,μ_1} denotes all the probability measures on $\mathcal{X} \times \mathcal{X}$ whose marginals according to the first and second variables are μ and μ_1 respectively, and $\mathcal{J}_\Pi$ is the set of pairs of function (f, g) such that $f(x) - g(y) \leq c(x, y)$ for all $x, y \in \mathcal{X}$ [14].

3. WORKING AT DENSITY LEVEL

The second approach we will consider consists in maximizing the likeliness of a probability model.

When it admits a density, learning the distribution μ_1 can be done through density estimation. Once the density is learned, it can be used to generate new samples with techniques such as Monte Carlo Markov Chain. Popular methods that go down this path are contrastive divergence and score matching, which learned the logarithm of the density based on

[2] E.g., $\mathcal{F}$ could be chosen as the image of a Stein operator [8].

[3] For example, if one considers μ_1 to be the uniform measure on $(0, 1/4) \cup (3/4, 1)$, $\mathcal{F}$ to be all the functions taking values in $[0, 1]$, and φ to parameterize the uniform distributions on $(a, a + 1/2)$ for $a \in [0, 1/2]$, there is no stable saddle point that the optimization can converge too. Indeed, if the adversary is $f = \mathbf{1}_{(1/2,1)}$, it will push φ to learn the uniform distribution on $(1/2, 1)$, which will lead the adversary f to become $\mathbf{1}_{(0,1/2)}$, which in turn will push φ to the uniform distribution there, creating an infinite loop.

[4] Seeing the characteristic function as the Fourier transform of a measure, it identifies with a negative Sobolev norm H^{-d-1}.

Kullback-Leibler and Fisher divergence respectively together with computational tricks [15]. However, this work is focused on learning pushforward maps.

Trackable pushforward density. In particular the map $\psi : \mathcal{X} \to \mathcal{X}$, that parametrizes $\mu = \psi_{\#}\mu_0$ from some base measure $\mu_0 \in \Delta_{\mathcal{X}}$,[5] can be learned using the change of variable formula

$$p(x) = p_0(\psi^{-1}(x))|\det(D\psi^{-1}(x))|. \tag{4}$$

and plugging $p(x)$ into some variational objective. Here, the densities p and p_0 formally corresponds to the Radon-Nikodym derivatives of μ and μ_0 respectively against a base measure, which is equally used to compute the Jacobian $D\psi$ of ψ,

$$p(x) := \frac{\mathrm{d}\mu}{\mathrm{d}x}(x), \quad p_0(x) := \frac{\mathrm{d}\mu_0}{\mathrm{d}x}(x).$$

Once the map ψ is learned, new samples are easily obtained as $\psi(X_0)$ for $X_0 \sim \mu_0$ (supposedly easy to generate). Following this perspective, [16] suggests an invertible architecture with easily computable Jacobian determinants to be trained with log-likelihood maximization. Composing many such mapping assimilates to a (normalizing) flow [17], which in the limit of composing infinitely many infinitesimal displacements can be seen as the solution of an ordinary differential equation [18].

Variational auto-encoder. Rather than providing new samples, $\varphi(z)$ could be used as the parameter of a well-known distribution from which to sample X, e.g., $(X|Z = z)$ is the Gaussian $\mathcal{N}(\varphi(z), 1)$. This creates a joint probability model on (X, Z) where the density $p(z)$ is fixed according to ν, and $p(x|z)$ is to be optimized to maximize the likelihood of the data. Computing the likelihood $p(x)$ implies a marginalization over z, which is usually too costly to compute. However, the evidence lower bound (ELBO), stating that[6]

$$\log p(x) = \max_{q_x \in \Delta_{\mathcal{Z}}} \mathbb{E}_{Z\sim q_x}[\log(\frac{p(x,Z)}{q_x(Z)})],$$

allows to derive a tractable objective to learn $\varphi : \mathcal{Z} \to \mathcal{X}$ by maximizing $\sum_i \mathbb{E}_{Z\sim q_{x_i}}[\log p(x_i, Z) - \log q_{x_i}(Z)]$ over both $q : \mathcal{X} \to \Delta_{\mathcal{Z}}$, seen as the encoder, and φ (which parameterizes $p(x|z)$), seen as the decoder [19] (see also [20]).

4. BUILDING MAPS FROM FLOWS

The third approach we will consider focuses on flows $t \mapsto \mu_t$ that transform continuously (with respect to some Wasserstein topology) the base measure μ_0 into the target measure μ_1.

Stochastic differential equations. In order to build a map that goes from a Gaussian variable $\mu_0 = \mathcal{N}(0, I)$ to μ_1, one could use a whitening process that map μ_1 to μ_0 before trying to invert it. Stochastic processes offer such constructions. E.g., one may initialize a random particle at $X_0 \sim \mu_1$, and diffuse it according to an energy potential $U_t(x) = \nabla E_t(x)$ that pulls the particle towards low energy regions, together with random noise that excites the particle in various random directions. Formally, the particle X_t is initialized at $X_0 \sim \mu_1$ and follows the stochastic differential process

$$\mathrm{d}X_t = -U_t(X_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t, \tag{5}$$

where B_t is the Brownian motion.[7] The specific case where $E_t(x) = \|x\|^2/2$ is known as the Ornstein-Uhlenbeck process, which maps $X_0 \sim \mu_1$ to $X_\infty \sim \mu_0 = \mathcal{N}(0, I)$, and could be reversed to go from $Z \sim \mu_0$ to $X \sim \mu_1$ through *stochastic simulation*. This is the basis of diffusion models [22], but how does it relate to the original problem of finding a *deterministic mapping* between μ_0 and μ_1? The answer leverages the flow $t \mapsto \mu_t$, defined through μ_t the law of $X_{(1-t)^{-1}}$.

From stochastic particle evolution to transport map. When a path that maps $t \in [0, 1]$ to $\rho_t \in \Delta_{\mathcal{X}}$ happens to be absolutely continuous with respect to the p-Wasserstein topology, Theorem 8.3.1 of [23] implies the existence of a velocity field $v_t \in L^p(\rho_t)$ that governs the evolution of ρ_t through the "continuity equation"

$$\mathrm{d}\rho_t = -\operatorname{div}(v_t\rho_t)\,\mathrm{d}t. \tag{6}$$

When ρ_t is the law of X_t that derives from equation (5), v is characterized by $v_t(x) = -U_t(x) - \nabla \log r_t(x)$, where $r_t(x)$ denotes the Lebesgue density of ρ in x. In essence, $v_t(x)$ characterizes the expected velocity of X_t when at $X_t = x$. Moreover, when $(t, x) \mapsto v_t(x)$ is bounded, Lipschitz continuous with respect to x, and continuous in t, Theorem 4.4 of [24] implies that the continuity equation (6) can be integrated with $\chi_t : \mathcal{X} \to \mathcal{X}$

$$\mathrm{d}\chi_t = v_t\,\mathrm{d}t, \tag{7}$$

from χ_0 being the identity to obtain a mapping $\chi = \chi_1$ such that $\chi_{\#}\rho_0 = \rho_1$.[8]

The need to speed up inference. Recently, practitioners have focused on neural networks that map (t, x) to $v_t(x)$. Such a network can be learned through score or flow matching [26]. They then use it at inference time to simulate the reverse stochastic process (5) leading from $Z \sim \mu_0$ to $X \sim \mu_1$, or to compute trajectories $t \mapsto \chi_{(1-t)^{-1}}(Z)$ by integration of the ODE (7). While diffusion models do not seem to be particularly more principled than other approaches, they are

[5] This parameterization ensures the Jacobian $D_z\psi$ to be a square matrix. If one uses $\mu = \varphi_{\#}\nu$ with $\varphi : \mathcal{Z} \to \mathcal{X}$, then one would have to consider the restriction of φ on the image of $D_z\varphi$, and $\mathrm{d}x$ to be a base measure on this subset to write the change of variable formula.

[6] This statement follows from that fact that for an arbitrary $q_x \in \Delta_{\mathcal{Z}}$, $\log p(x) = \mathbb{E}_{z\sim q_x}[\log(\frac{p(x,z)}{p(z|x)})] = \mathbb{E}_z[\log(\frac{p(x,z)}{q_x(z)})] + D_{KL}(q_x\|p(z|x))$, The maximum is reached for $q_x = p(z|x)$.

[7] Inspired by physical systems, one could enrich equation (5) to describe particle interactions, see e.g. [21]. While it has not been explored much in the literature, it may help to engineer better flow, or to enforce sample diversities.

[8] Interestingly, many flow maps provide optimal transport plans [25].

performing remarkably well on images in practice. This may be because $(t, x) \mapsto v_t(x)$ is expected to be regular, hence easy to learn, especially with some adapted U-net architecture; and because the error between trajectories obtained by integration of an estimated flow $\hat{v}_t$ and integration of the real one v_t is relatively well-behaved (see, e.g., Section 2.4 in [27]).

However, generating new samples with flow-based models currently relies on expensive stochastic simulation or flow integration at inference time. Some attempts have been made to learn straight flows that are fast to integrate [28]. Others have tried to globally integrate the velocity v_t into a map ψ, which has been referred to as distillation, as one want to "distill" the neural network that parameterized $(t, x) \mapsto v_t(x)$ into a new neural network parameterizing ψ [29]. But why not directly learn the pushforward $\psi = \chi_\infty^{-1}$ rather than going through the convoluted construction of learning and integration of flows?

From network to optimization architecture. Let us now focus on the following research question: GANs are known to be hard and unstable to train, but could diffusion models help us engineer better optimization procedures? Before training, a GAN (or an auto-encoder) will more or less map Gaussian noise, seen as the distribution μ_0, to Gaussian noise, seen as the distribution ρ_∞. Eventually, one could fine-tune it iteratively to map μ_0 to ρ_t for t decreasing during the different rounds of fine tuning. At the end, one would obtain a mapping from $\mu_0 = \rho_\infty$ to $\mu_1 = \rho_0$. The fine-tuning would always employ the same loss at each iteration, but the data would be modified little by little, starting with really noisy images, i.e., $X \sim \rho_t$ for a big t, to sharper high-resolution images, i.e., $X \sim \rho_t$ for a small t.

Diffusion models have seduced theorists, they may wonder what our suggestion corresponds to in more abstract terms. Extending on the seminal papers of Otto [30, 31], one can show that the flow $t \mapsto \rho_t$ that solves (5) for a time-independent potential $U_t = \nabla E$ corresponds to the gradient flow (for some topology) of the objective D_2 (3) for $\mathcal{F} = \{f_i | i \in \mathbb{N}\}$ and $\tau(f_i) \propto \lambda_i$, where (λ_i, f_i) is the eigendecomposition of the operator representing the quadratic form $f \mapsto \langle \|\nabla f\|^2 \,|\, \mu_0 \rangle$ in $L^2(\mu_1)$ where $\mu_1 \propto \exp(-E(x))\,\mathrm{d}x$.[9] In other terms, denoising diffusion probability models that learn v_t through the Ornstein-Uhlenbeck process so that the base sampling distribution $\mu_0(\mathrm{d}x) \propto \exp(-x^2)$ is a Gaussian distribution are implicitly following and reverting a gradient flow. If diffusion models encode the full flow $(t, x) \mapsto v_t(x)$ (6) in their architecture, we are suggesting to rather encode this flow into an optimization scheme to learn a network f_θ so that after t "epochs" of optimization $f_{\theta_t} = \psi_t^{-1}$ (7), and at the end of the optimization $(f_{\theta_\infty})_\# \mu_0 = \mu_1$.

[9] It is relatively standard to show that $\mathrm{d}\langle f_i \,|\, \rho_t \rangle = -\lambda_i \langle f_i \,|\, \rho_t \rangle \,\mathrm{d}t$ for all $i \in \mathbb{N}$ and that $f_0 = 1$ [32]. As a consequence, $\langle f_i \,|\, \rho_\infty \rangle = \langle f_i \,|\, \mu_0 \rangle = 0$, and $\frac{\mathrm{d}\rho_t}{\mathrm{d}t} = -\sum_i \lambda_i \langle f_i \,|\, \rho_t \rangle f_i^* = -\frac{\mathrm{d}}{\mathrm{d}\rho} \sum_i \lambda_i \langle f_i \,|\, \rho_t \rangle^2$, which assimilates to $\mathrm{d}D_2(\rho_t, \mu_0; \tau)/\,\mathrm{d}\rho$. Details to define the different topologies (to take adjoints and gradients) and prove formally this result are beyond the scope of this paper.

5. CONCLUSION

There exists many perspectives on sampling, even when we restrict ourselves to the search of a pushforward map while only accessing samples from the target distribution.

A straightforward approach consists in defining losses using test functions to quantify differences between measures, as explained in Section 2. Among the methods based on this principle, GANs have provided the most exciting results, but their training is hard, requiring skilled engineering.

This has led to the rise of another stream of research based on density-centered considerations, as detailed in Section 3. One can e.g. learn pushforward maps by maximizing probability likelihood, though it involves ensuring mapping invertibility and computing Jacobian determinants, which is constraining. Moreover, the target distribution μ_1 might not have a density, which is notably the case when the data span uniformly a manifold in the input space. This hurdle can be avoided by artificially injecting noise to the data to create a distribution μ_1' admitting a density before adding a denoising step to recover μ_1 from μ_1'. Several model evolutions have led to cascading many noise injection and denoising processes, which were ultimately understood through the lens of Brownian motion and reversible diffusion models, explained in Section 4. Those diffusion models are currently state-of-the-art, although they require extensive simulation to generate new samples. Remarkably, they are implicitly associated with pushforward maps.

Getting a neural network to efficiently learn such a pushforward map would be a major advance for generative AI. Another, perhaps less discussed issue is that the target measure is only accessed through a finite number of samples, and rules to extrapolate from them arise mainly as a consequence of the biases induced by network architectures and optimization schemes–more principledness would be desirable.

This paper reviewed several principles to learn pushforward maps. Future work consists in experimenting with those to extract more practical knowledge for generative AI across different modalities.

Acknowledgment. VC would like to thank Brandom Amos, Alberto Bietti, Florian Bordes, Léon Bottou, Ricky Chen, Valentin de Bortoli, Carles Domingo-Enrich, Yann LeCun, Loucas Pillaud-Vivien, Aram Pooladian and Yann Olivier for useful discussions.

[1] The Lancet Regional Health – Europe, "Embracing generative ai in health care," *The Lancet Regional Health - Europe*, 2023.

[2] Matt Novak, "Artist reveals his award-winning 'photo' was created using AI," *Forbes*, 2023.

[3] OpenAI, "GPT-4 technical report," Tech. Rep., OpenAI, 2023.

[4] McKinsey & Company, "The economic potential of generative AI: The next productivity frontier," Tech. Rep., McKinsey & Company, 2023.

[5] Alfred Müller, "Integral probability metrics and their generating classes of functions," *Advances in Applied Probability*, 1997.

[6] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alexander Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, 2012.

[7] Martin Arjovsky, Soumith Chintala, and Léon Bottou, "Wasserstein GAN," 2017.

[8] Jackson Gorham and Lester Mackey, "Measuring sample quality with Stein's method," in *Advances in Neural Information Processing Systems*, 2015.

[9] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *International Conference on Learning Representations*, 2016.

[10] Gábor Székely and Maria Rizzo, "Energy statistics: A class of statistics based on distances," *Journal of Statistical Planning and Inference*, 2013.

[11] Yury Polyanskiy and Wu Yihong, *Information Theory: From Coding to Learning*, Cambridge University Press, 2022.

[12] Svetlozar Rachev, Lev Klebanov, and Stoyan Stoyanov andFrank Fabozzi, *The Methods of Distances in the Theory of Probability and Statistics*, Springer, 2013.

[13] Leon Bottou, Martin Arjovsky, David Lopez-Paz, and Maxime Oquab, "Geometrical insights for implicit generative modeling," in *Braverman Readings in Machine Learning: Key Ideas from Inception to Current State*. Springer, 2018.

[14] Cédric Villani, *Optimal Transport: Old and New*, Springer, 2009.

[15] Yang Song and Diederik P. Kingma, "How to train your energy-based models," 2021.

[16] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio, "Density estimation using real NVP," in *International Conference on Learning Representations*, 2017.

[17] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," *Journal of Machine Learning Research*, 2021.

[18] Ricky Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud, "Neural ordinary differential equations," in *Advances in Neural Information Processing Systems*, 2019.

[19] Diederik Kingma and Max Welling, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, 2019.

[20] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov, "Importance weighted autoencoders," in *International Conference on Learning Representations*, 2016.

[21] Sylvia Serfaty, "Systems of points with coulomb interactions," in *Proceedings of the International Congress of Mathematicians*, 2018.

[22] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021.

[23] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré, *Gradient Flows in Metric Spaces and in the Space of Probability Measures*, Springer, 2005.

[24] Filippo Santambrogio, *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*, Springer, 2015.

[25] Valentin Khrulkov, Gleb Ryzhakov, Andrei Chertkov, and Ivan Oseledets, "Understanding ddpm latent codes through optimal transport," in *International Conference on Learning Representations*, 2023.

[26] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le, "Flow matching for generative modeling," in *International Conference on Learning Representations*, 2023.

[27] Michael Albergo, Nicholas Boffi, and Eric Vanden-Eijnden, "Stochastic interpolants: A unifying framework for flows and diffusions," 2023.

[28] Xingchao Liu, Chengyue Gong, and Qiang Liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," in *International Conference on Learning Representations*, 2023.

[29] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans, "On distillation of guided diffusion models," in *Conference on Computer Vision and Pattern Recognitio*, 2023.

[30] Richard Jordan, David Kinderlehrer, and Felix Otto, "The variational formulation of the Fokker-Planck equation," *Journal of Mathematical Analysis*, 1998.

[31] Felix Otto, "The geometry of dissipative evolution equations the porous medium equation," *Communications in Partial Differential Equations*, 2001.

[32] Dominique Bakry, Ivan Gentil, and Michel Ledoux, *Analysis and Geometry of Markov Diffusion Operators*, Springer, 2014.