Wasserstein proximal operators describe score-based generative models and resolve memorization

Benjamin J. Zhang[1], Siting Liu[2], Wuchen Li[3], Markos A. Katsoulakis[1], and Stanley J. Osher[2]

[1]Department of Mathematics and Statistics, University of Massachusetts Amherst
[2]Department of Mathematics, University of California, Los Angeles
[3]Department of Mathematics, University of South Carolina, Columbia

February 9, 2024

Abstract

We focus on the fundamental mathematical structure of score-based generative models (SGMs). We first formulate SGMs in terms of the Wasserstein proximal operator (WPO) and demonstrate that, via mean-field games (MFGs), the WPO formulation reveals mathematical structure that describes the inductive bias of diffusion and score-based models. In particular, MFGs yield optimality conditions in the form of a pair of coupled partial differential equations: a forward-controlled Fokker-Planck (FP) equation, and a backward Hamilton-Jacobi-Bellman (HJB) equation. Via a Cole-Hopf transformation and taking advantage of the fact that the cross-entropy can be related to a linear functional of the density, we show that the HJB equation is an uncontrolled FP equation. Second, with the mathematical structure at hand, we present an *interpretable kernel-based* model for the score function which dramatically improves the performance of SGMs in terms of training samples and training time. In addition, the WPO-informed kernel model is explicitly constructed to avoid the recently studied memorization effects of score-based generative models. The mathematical form of the new kernel-based models in combination with the use of the terminal condition of the MFG reveals new explanations for the manifold learning and generalization properties of SGMs, and provides a resolution to their memorization effects. Finally, our mathematically informed, interpretable kernel-based model suggests new scalable bespoke neural network architectures for high-dimensional applications.

Key words. Score-based generative models; Wasserstein proximal operators; Memorization; Manifold learning; Kernel methods; Cole–Hopf transformation; Reverse heat equation

1 Introduction

Proximal operators are powerful tools in optimization [1]. *Wasserstein* proximal operators (WPO) have been found to be crucial tools for formulating flow-based generative models through connections made by Hamilton-Jacobi equations [2] and mean-field games [3, 4]. This paper formulates score-based generative models (SGMs) [5] and related denoising diffusion models [6] with the Wasserstein proximal operator. We show that SGMs can be concisely described in terms of the Wasserstein proximal operator of *cross-entropy*, which places the method in the context of other generative flows and provides new pathways for analysis of SGMs. For example, the manifold learning aspects of proximal operators on 1-Wasserstein space have been noted in [7, 8]; the Wasserstein proximal operator therefore provides an additional justification for the manifold learning properties of SGMs [9].

M. Katsoulakis and B. Zhang are partially funded by AFOSR grant FA9550-21-1-0354. M.K. is partially funded by NSF DMS-2307115 and NSF TRIPODS CISE-1934846. S. Liu and S. Osher are partially funded by AFOSR MURI FA9550-18-502 and ONR N00014-20-1-2787. W. Li is supported by AFOSR MURI FP 9550-18-1-502, AFOSR YIP award No. FA9550-23-1-0087, NSF DMS-2245097, and NSF RTG: 2038080.

The empirical success of SGMs has induced much research into their statistical properties and mathematical foundations [9, 10, 11]. There is much interest in how to construct better models for SGMs that train faster with less data. Some examples include designing critically damped diffusions that converge to Gaussians faster [12], finding the optimal time horizon [13], implementing high order SDE integrators [14], or distilling diffusion models into consistency models [15]. More interestingly, there is a growing body of work documenting so-called *memorization effects* of diffusion models [16, 17, 18, 19, 20]. In particular, [16] presents an example where learning a score function well with the denoising score-matching objective produces a generative model that is equivalent to a kernel density estimate. Moreover, they demonstrated that the primary generative capabilities of the resulting model were due to early stopping, which produced samples equivalent to noisy replicas of the training data. Furthermore, this memorization effect has even been noted in major newspapers [21], in which they demonstrated that a popular score-based text-to-image generator, Midjourney, may output mildly modified copyrighted images that were likely part of its training data. Copyright infringement is a broader, systemic challenge for generative modeling that may entangle engineers and practitioners in litigation [22].

The relationship between Wasserstein proximal operators and generative modeling has previously been explored in gradient flows [23] and in generative adversarial networks [24]. We present a fundamental characterization of SGMs in terms of the Wasserstein proximal operator which yields an *interpretable* kernel-based approach to approximating score functions. In particular, via kernel formulas for computing regularized Wasserstein proximal operators [4] and their relation to Hamilton-Jacobi-Bellman (HJB) equations, we construct a *WPO-informed* kernel model for the score function using the Green's functions related to the HJB equation. Moreover, we demonstrate that enforcing the terminal condition of the associated HJB equation to learn the parameters of the kernel-based formula is tantamount to learning the manifold on which the data lies. The combination of the mathematical form of the WPO-kernel model and its training procedure provides a resolution to the memorization effects of SGMs. Empirical evidence demonstrate that the mathematical structure encoded by WPO-informed kernel model exhibits faster learning of the score function with less data and produces a SGM that generalizes. We discuss our contributions in the following two sections.

Contribution 1: Formulating score-based generative models through Wasserstein proximal operators. In Section 3, we provide a fundamental characterization of score-based generative model as Wasserstein proximal operators of cross-entropy. The mean-field games (MFG) formulation of SGMs and regularized WPOs established in [3] and [4], respectively, establish the connection between SGMs and WPOs. The optimality conditions of the SGM MFG yield a pair of PDEs: a *forward* controlled Fokker-Planck (FP) equation, and a *backward* Hamilton-Jacobi-Bellman (HJB) equation. Moreover it was shown in [3] that the implicit score-matching objective is a direct consequence of the MFG formulation, and shows that the ISM objective is related to simultaneously minimizing the Wasserstein metric and the cross-entropy loss.

Our primary contribution in this section is connecting the Wasserstein proximal operator of linear energies [4] to the MFG perspective of SGMs [3]. Kernel formulas for approximating the Wasserstein proximal operator in [4], and this connection allows for the construction of kernel-based models for the score function. We will review how the particular form of the MFG, specifically the inclusion of the cross-entropy as the terminal cost decouples the set of PDEs, implying that the HJB equation alone characterizes SGM. Moreover, by a Cole-Hopf transformation, the HJB equation is equivalent to an uncontrolled FP equation [25, 3]. The MFG's *forward-backward* structure explains the noising-denoising nature of score-based generative models. Establishing the connection between WPOs and SGMs justifies the use of kernel formulas from [4] to provide a closed form solution to the score function, which is the topic of the next section.

We also highlight an observation that the backward-forward PDE system which arises from the SGM MFG provides an alternative perspective for reversing the heat equation. Solving the reverse heat equation is an ill-posed problem in numerical PDEs. By applying PDE theory (mean-field games) to the SGM system, we discuss how the one-way coupled PDE system corresponding to SGMs yields a well-posed system with an associated optimization problem that has the effect of reversing the heat equation, provided that some minimal information about the initial condition is available.

Contribution 2: Deconstructing score-based models with an interpretable WPO-informed kernel model that *resolves memorization and generalizes.* With the WPO formulation of SGM at hand, our main contribution in Section 4 is the WPO-informed kernel model which is an *interpretable* model that unravels the mathematical structure of the score function. The WPO formulation of SGM provides a description of the *inductive bias* of SGM, that is, the inherent mathematical structure of the score function instantiated as the gradient of the solution of the HJB equation (21). The solution of the HJB equation can be written in closed form as the Cole-Hopf transform of a kernel model. Incorporating the structure of the HJB equation allows the WPO-informed kernel score model to be trained quickly with less data. In Section 5, we demonstrate the advantages of the kernel formula in illustrative numerical examples. The kernel-based model in (25) is the main result of the WPO formulation of SGMs, which we preview here

$$\hat{\pi}_\theta(x; \{Z_i\}_{i=1}^N) = \frac{1}{N}\sum_{i=1}^N \frac{\det \mathbf{\Gamma}_\theta(Z_i)}{(2\pi)^{d/2}} \exp\left(-\frac{(x-Z_i)^\top \mathbf{\Gamma}_\theta(Z_i)(x-Z_i)}{2}\right).$$

The kernel in the WPO-informed model is informed by the *Green's function* of the HJB equation, which comes from the optimality conditions of the MFG. The WPO-informed kernel model is a linear combination of Green's function centered on the training data, which solves the HJB equation exactly by construction. Given training data $\{Z_i\}_{i=1}^{N_{\text{train}}}$ from data distribution π we learn the precision $\mathbf{\Gamma}_\theta(Z_i)$ (inverse covariance matrices) around each of the kernel centers, which are a subset of the training data[1]. It is simply an extension of the kernel formulas for approximating the Wasserstein proximal operators of linear operators first derived in [4]. Ostensibly this model is simply a Gaussian mixture model. The view that SGMs are GMMs or kernel formulas has been noted recently in [16], which further supports the use of the WPO-kernel model. The WPO formulation of SGM (16) and its solution described by Proposition 3.1 provides a kernel representation formula that encodes the inductive bias of SGMs. A Gaussian mixture model is a natural choice for expressing the inductive bias as approximating the kernel representation formula (20) admits a kernel score model that can be computed in closed form. This fact is presented in Proposition 4.1.

A key difference from a standard GMM is that the model parameters θ are trained via implicit score-matching only at the terminal condition. As the kernel-based formula solves the HJB equation exactly except at the terminal condition, there is no need to perform score-matching for any other time other than the terminal time (see Proposition 4.1). The local precision matrices are learned by minimizing the implicit score-matching objective, which has the effective of learning the underlying Riemannian manifold on which the data distribution lies [26]. This allows the resulting generative model to generalize and avoid memorization effects.

The resulting model provides an explainable, interpretable formulation of score-based generative model grounded through interconnections among MFGs, information theory, optimal transport, manifold learning, and optimization. Our model clarifies the effectiveness of *early stopping* and manifold learning properties of SGMs. Moreover, the manifold learning properties may mathematically explain the benefits of latent diffusion methods [27, 28], where manifold learning is treated separately from the SGM.

2 Background on score-based generative models and Wasserstein proximal operators

We review score-based generative models with SDEs and kernel approximation to Wasserstein proximal operators of linear energies as presented in [5] and [4], respectively. In Section 3 we show how the two seemingly disparate topics are mathematically equivalent.

Notation. Let $\mathcal{P}_2(\mathbb{R}^d)$ denote the space of probability distributions with finite second moments. The pair $(\mathcal{P}_2(\mathbb{R}^d), \mathcal{W}_2)$ is a complete separable metric space on $\mathcal{P}_2(\mathbb{R}^d)$, where $\mathcal{W}_2(\rho,\rho')$ is the 2-Wasserstein distance [29]. The density functions of probability distributions $\rho \in \mathcal{P}_2(\mathbb{R}^d)$ are denoted $\rho(x)$.

2.1 Score-based generative models

Suppose we wish to produce samples from target data distribution $\pi \in \mathcal{P}_2(\mathbb{R}^d)$ given only a finite number of samples $\{X_i\}_{i=1}^N \sim \pi$. Let $f : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ be a vector field for some scalar $T > 0$, and let $\beta(t)$ be a

[1] In numerical experiments, we chose a subset of training data to be the kernel centers, but in general this is not necessary.

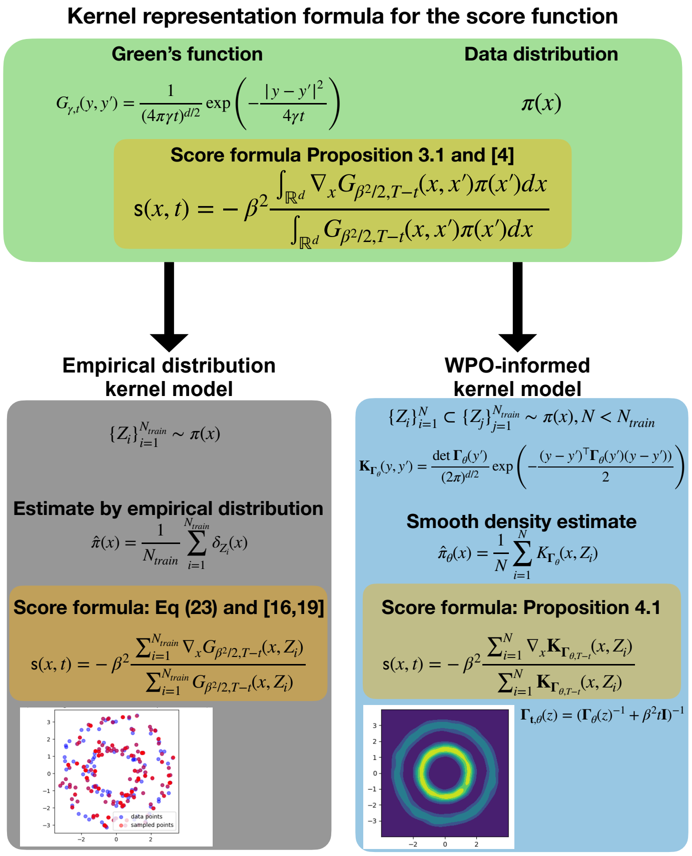

Kernel representation formula for the score function
Green's function
$G_{\gamma,t}(y,y') = \frac{1}{(4\pi\gamma t)^{d/2}} \exp\left(-\frac{|y-y'|^2}{4\gamma t}\right)$
Data distribution
$\pi(x)$
Score formula Proposition 3.1 and [4]
$\mathsf{s}(x,t) = -\beta^2 \frac{\int_{\mathbb{R}^d} \nabla_x G_{\beta^2/2,T-t}(x,x')\pi(x')dx}{\int_{\mathbb{R}^d} G_{\beta^2/2,T-t}(x,x')\pi(x')dx}$
Empirical distribution kernel model
$\{Z_i\}_{i=1}^{N_{train}} \sim \pi(x)$
Estimate by empirical distribution
$\hat{\pi}(x) = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} \delta_{Z_i}(x)$
Score formula: Eq (23) and [16,19]
$\mathsf{s}(x,t) = -\beta^2 \frac{\sum_{i=1}^{N_{train}} \nabla_x G_{\beta^2/2,T-t}(x,Z_i)}{\sum_{i=1}^{N_{train}} G_{\beta^2/2,T-t}(x,Z_i)}$
data points
sampled points
WPO-informed kernel model
$\{Z_i\}_{i=1}^{N} \subset \{Z_j\}_{j=1}^{N_{train}} \sim \pi(x), N < N_{train}$
$\mathbf{K}_{\Gamma_\theta}(y,y') = \frac{\det \mathbf{\Gamma}_\theta(y')}{(2\pi)^{d/2}} \exp\left(-\frac{(y-y')^{\mathsf{T}}\mathbf{\Gamma}_\theta(y')(y-y')}{2}\right)$
Smooth density estimate
$\hat{\pi}_\theta(x) = \frac{1}{N}\sum_{i=1}^{N} K_{\Gamma_\theta}(x,Z_i)$
Score formula: Proposition 4.1
$\mathsf{s}(x,t) = -\beta^2 \frac{\sum_{i=1}^{N} \nabla_x \mathbf{K}_{\Gamma_{\theta,T-t}}(x,Z_i)}{\sum_{i=1}^{N} \mathbf{K}_{\Gamma_{\theta,T-t}}(x,Z_i)}$
$\mathbf{\Gamma}_{t,\theta}(z) = (\mathbf{\Gamma}_\theta(z)^{-1} + \beta^2 t\mathbf{I})^{-1}$

Figure 1: The core idea of this paper is that since the score function has a kernel representation formula, approximations to the score should respect the structure of the formula. Use of the empirical distribution to construct a kernel formula for the score function memorizes the training data. Our WPO-informed kernel model learns local precision matrices via the terminal condition of a HJB equation, which produces a kernel-based model that generalizes better and exhibits manifold learning.

positive function for $t \in [0,T]$. Let $Y(s)$ be the stochastic process evolving according to SDE

$$\begin{aligned} \mathrm{d}Y(s) &= -f(Y(s), T-s)\,\mathrm{d}s + \beta(T-s)\,\mathrm{d}W(s) \\ Y(0) &\sim \pi \end{aligned} \tag{1}$$

where $Y(s) \sim \eta(\cdot, s)$ and $W(s)$ is a standard Brownian motion. This process adds noise to samples from π; typically f and β are chosen so that π is approximately normal for sufficiently large T. Score-based generative models aim to learn a controlled SDE that reverses the evolution of the noising process. In [30] it was first found that the process $X(t) \sim \rho(\cdot, t)$

$$\begin{aligned} \mathrm{d}X(t) &= \left[f(X(t),t) + \beta(t)^2 \nabla \log \eta(X(t), T-t)\right]\,\mathrm{d}t + \beta(t)\,\mathrm{d}W(t) \\ X(0) &\sim \eta(\cdot, T) \end{aligned} \tag{2}$$

reverses the evolution of $Y(s)$ with $\rho(\cdot,t) = \eta(\cdot, T-t)$ so that $\rho(\cdot,0) = \eta(\cdot,T)$ and $\rho(\cdot,T) = \pi(\cdot)$. Assuming the score function $\mathsf{s}(y,s) = \nabla \log \eta(y,s)$ is learned, then new samples from π can be obtained by evolving samples from $\eta(\cdot,T)$ (which are approximately normal) through the SDE of $X(t)$. To learn the score function, [5] trains a neural net $\mathsf{s}_\theta : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ by minimizing a score-matching loss function. The explicit score-matching objective (ESM)

$$C_{ESM}(\theta) = \int_0^T \beta(T-s)^2 \mathbb{E}_{\eta(\cdot,s)} \left[\|\mathsf{s}_\theta(Y(s),s) - \nabla \log \eta(Y(s),s)\|^2 \right] \, \mathrm{d}s \tag{3}$$

is generally intractable to compute since evaluations of the density function are not available [31]. One of the practical alternatives is the implicit score-matching objective (ISM) [31]

$$C_{ISM}(\theta) = \int_0^T \beta(T-s)^2 \mathbb{E}_{\eta(\cdot,s)} \left[\|\mathsf{s}_\theta(y,s)\|^2 + 2\nabla \cdot \mathsf{s}_\theta(y,s) \right] \, \mathrm{d}s \tag{4}$$

which applies integration-by-parts to avoid direct evaluations of the score function. Here $\nabla\cdot$ denotes the divergence operator. Another loss function is the denoising score-matching (DSM) objective [5, 6, 32]

$$C_{DSM}(\theta) = \int_0^T \beta(T-s)^2 \mathbb{E}_{Y_0 \sim \pi} \mathbb{E}_{Y(s) \sim \eta(\cdot, s|Y_0)} \left[\|\mathsf{s}_\theta(Y(s),s) - \nabla \log \eta(\cdot, s|Y_0)\|^2 \right] \, \mathrm{d}s, \tag{5}$$

which requires knowledge of the conditional score function $\nabla \log \eta(y, s|Y_0)$. The functions f and β are typically chosen so that the conditional score function is known in closed form. The DSM objective is most frequently used in practical applications as it does not require gradient evaluations when the noising process is a linear SDE. While these three objective functions are equivalent in the limit of infinite data, i.e., they share the same minimizers, the properties of their empirical losses are not fully understood.

2.2 Kernel formulas for regularized Wasserstein proximal operators

Proximal operators often arise in the optimization of nonconvex and nonsmooth functions [1]. The Wasserstein proximal operator (WPO) is the instantiation of the proximal operator on the space of probability distributions with finite second moments $\mathcal{P}_2(\mathbb{R}^d)$. The WPO is a mapping on the Wasserstein space which is a metric space on $\mathcal{P}_2(\mathbb{R}^d)$ along with the 2-Wasserstein distance [29]. Intuitively, given some functional on Wasserstein space (e.g., a probability divergence with respect to some fixed distribution), one evaluation of the proximal operator can be interpreted as a single step of gradient descent for that functional on Wasserstein space.

Given a potential energy function $V : \mathbb{R}^d \to \mathbb{R}$, a linear functional on Wasserstein space is

$$\mathcal{V}(\rho) = \int V(x)\rho(x) \, \mathrm{d}x. \tag{6}$$

Often, the potential energy function is the negative log density of some target density, i.e., $V(x) = -\log \pi(x)$. The Wasserstein proximal operator of $\mathcal{V}$ is a mapping on $\mathcal{P}_2(\mathbb{R}^d)$

$$\rho_h = \mathrm{WProx}_{h\mathcal{V}}(\rho_0) \coloneqq \operatorname*{arg\,min}_{\rho \in \mathcal{P}_2(\mathbb{R}^d)} \mathcal{V}(\rho) + \frac{\mathcal{W}_2(\rho_0, \rho)^2}{2h}, \tag{7}$$

where $h > 0$ is a scalar constant. The WPO has a dynamic formulation via the Benamou-Brenier formula [29]: for any $T > 0$,

$$\frac{\mathcal{W}_2^2(\rho_0, \mu)}{2h} = \inf_v \left\{ \int_0^h \int_{\mathbb{R}^d} \frac{1}{2} |v(x,t)|^2 \rho(x,t) \, \mathrm{d}x \, \mathrm{d}t : \partial_t \rho + \nabla \cdot (v\rho) = 0, \rho(x,0) = \rho_0(x), \rho(x,h) = \mu(x) \right\}. \tag{8}$$

The WPO is, therefore, associated with the potential mean-field game

$$\begin{aligned} &\min_{\rho, v} \left\{ -\int_{\mathbb{R}^d} V(x)\rho(x,h) \, \mathrm{d}x + \int_0^T \int_{\mathbb{R}^d} \frac{1}{2} |v(x,t)|^2 \rho(x,t) \, \mathrm{d}x \, \mathrm{d}t \right\} \qquad (9) \\ &\text{s.t. } \frac{\partial \rho}{\partial t} + \nabla \cdot (v\rho) = 0, \ \rho(x,0) = \rho_0(x), \end{aligned}$$

which has optimality conditions in the form of two coupled nonlinear partial differential equations (PDEs)

$$\begin{cases} -\dfrac{\partial U}{\partial t} + \dfrac{1}{2}\left|\nabla U\right|^2 = 0,\, U(x,h) = V(x) \\ \dfrac{\partial \rho}{\partial t} - \nabla \cdot (\rho \nabla U) = 0,\, \rho(x,0) = \rho_0(x). \end{cases} \tag{10}$$

The first equation is a Hamilton-Jacobi equation, while the second is simply the continuity equation with $v = -\nabla U$. The output of the WPO operator applied to the input distribution ρ_0 is the density function $\rho(x,T)$. Solving such a system in general requires numerical approximation; however, standard strategies for solving PDEs are quite intractable in high dimensions. To address the curse of dimensionality, [4] proposed the *regularized* Wasserstein proximal operator by introducing second derivative terms in the optimality conditions, which can be interpreted as a form of *entropic* regularization, similar to the entropic formulation of optimal transport [33]. Doing so allows using kernel formulas which provides a closed form solution to the resulting mean-field game. The regularized Wasserstein proximal operator with entropic regularization parameter $\gamma > 0$, $\rho_T(x) = \mathrm{WProx}_{T\mathcal{V},\gamma}(\rho_0)$, is associated with the PDE system

$$\begin{cases} -\dfrac{\partial U}{\partial t} + \dfrac{1}{2}\left|\nabla U\right|^2 = \gamma \Delta U,\, U(x,T) = V(x) \\ \dfrac{\partial \rho}{\partial t} - \nabla \cdot (\rho \nabla U) = \gamma \Delta \rho,\, \rho(x,0) = \rho_0(x), \end{cases} \tag{11}$$

where $\rho_T = \rho(x,T)$. In [4], it was shown that via a Cole-Hopf transform, the HJB equation in (11) is equivalent to the heat equation, which can be solved via Green's functions. Recall that the *heat kernel*, $G_{\gamma,t}(y,y')$, for diffusion constant γ is the *Green's function* of the heat equation and the convolution operator $*$ for a function $\phi : \mathbb{R}^d \to \mathbb{R}$ are defined

$$G_{\gamma,t}(y,y') = \frac{1}{(4\pi\gamma t)^{d/2}} \exp\left(-\frac{|y-y'|^2}{4\gamma t}\right),\; (G_{\gamma,t} * \phi)(x) = \int_{\mathbb{R}^d} G_{\gamma,t}(x,x')\phi(x')\,\mathrm{d}x'. \tag{12}$$

The solution to the PDE system (11) can therefore be expressed as

$$U(x,t) = -2\gamma \log\left(G_{\gamma,T-t} * e^{-\frac{V(x)}{2\gamma}}\right). \tag{13}$$

3 Deriving score-based generative models from Wasserstein proximal operators

In this section we show how score-based generative models can be formulated in terms of a regularized Wasserstein proximal operator of cross-entropy. Minimizing the cross-entropy functional over spaces of probability distributions is a frequent task in machine learning [34, 35]. We will show that score-based generative models produces samples of a single application of the Wasserstein proximal of the cross-entropy functional. Using results from [3], we derive canonical formulations score-based generative models as presented in [5].

3.1 Deriving SGM from regularized proximal operators of cross-entropy

Let $\pi \in \mathcal{P}_2(\mathbb{R}^d)$ be the target data distribution as defined in Section 2.1 and let $\mu \in \mathcal{P}_2(\mathbb{R}^d)$ be some arbitrary distribution. The cross-entropy of a distribution π with respect to μ is defined

$$\mathcal{H}(\mu,\pi) \coloneqq -\mathbb{E}_\mu\left[\log \pi(x)\right] = -\int_{\mathbb{R}^d} \mu(x) \log \pi(x)\,\mathrm{d}x. \tag{14}$$

The cross-entropy method is related to the Kullback-Leibler divergence and the entropy $\mathcal{E}(\mu) = -\mathbb{E}_\mu[\log \mu(x)]$ by the formula $\mathcal{D}_{KL}(\mu\|\pi) = -\mathcal{E}(\mu) + \mathcal{H}(\mu,\pi)$. Minimizing the cross-entropy appears frequently in numerous machine learning tasks [34, 35]; we will show that it also plays an interesting role in score-based generative models.

Starting from the regularized WPO mean-field game in (11) and results from [3], we show that SGMs can be fundamentally understood in terms of the cross-entropy loss and Wasserstein proximal operator. In particular, through a judicious choice of the regularization parameters γ and h, we will show that (11) is equivalent to the SGM formulation in Section 2.1. First, consider the Wasserstein proximal operator in (15) for $h = \beta^2 T$ for $\beta, T > 0$, and note its equivalence to

$$\mathrm{WProx}_{h\mathcal{H}}(\rho_0) = \underset{\rho \in \mathcal{P}_2(\mathbb{R}^d)}{\arg\min}\, \beta^2 \mathcal{H}(\rho, \pi) + \frac{\mathcal{W}_2(\rho_0, \rho)^2}{2T}. \tag{15}$$

Next, by choosing the entropic regularization parameter $\gamma = \beta^2/2$, notice that we obtain the mean-field game optimality conditions for the regularized WPO, $\mathrm{WProx}_{\beta^2 T\mathcal{H}, \beta^2/2}(\rho_0)$,

$$\begin{cases} -\dfrac{\partial U}{\partial t} + \dfrac{1}{2}|\nabla U|^2 = \dfrac{\beta^2}{2}\Delta U, \quad U(x,T) = -\beta^2 \log \pi(x) \\ \dfrac{\partial \rho}{\partial t} - \nabla \cdot (\rho \nabla U) = \dfrac{\beta^2}{2}\Delta \rho, \quad \rho(x,0) = \rho_0(x) \end{cases} \tag{16}$$

Combining the Cole-Hopf transformation with a time reparametrization, we apply the variable transformation $U(x,t) = -\beta^2 \log \eta(x, T-t)$ to the Hamilton-Jacobi equation and obtain the system

$$\begin{cases} \dfrac{\partial \eta}{\partial s} = \dfrac{\beta^2}{2}\Delta \eta, \; \eta(y,0) = \pi(y) \\ \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \beta^2 \nabla \log \eta) = \dfrac{\beta^2}{2}\Delta \rho, \quad \rho(x,0) = \rho_0(x) \end{cases} \tag{17}$$

The first equation is the heat equation, which in terms of SDEs, corresponds with an uncontrolled Brownian motion with diffusion coefficient β. The second equation is a controlled Fokker-Planck equation, where the controller is determined by the score function of the uncontrolled Brownian motion. Notice that these are precisely the noising (1) and denoising systems (2) described in Section 2.1 for $f \coloneqq 0$. Furthermore, if $\rho_0(x)$ is chosen to be the density $\eta(\cdot, T)$ at $s = T$, then ([3], Theorem 4.1) showed that the controlled Fokker-Planck will reverse the evolution of the heat equation. To solve the system of PDEs, one needs to find the solution to the HJB equation, whose gradient is precisely the score function. Therefore, we have shown that the pair of noising and denoising SDEs that appear in score-based generative models are naturally encoded in the backward-forward structure of the mean-field game representation of the regularized Wasserstein proximal operator.

Moreover, in ([3], Theorem 4.2), it was shown that the mean-field game in (9) can be directly related to the implicit score matching objective in (4), meaning that optimizing the ISM and learning the score is equivalent to solving the MFG (16). With these connections, score-based generative models can be summarized with the forward and inverse WProx notation:

$$\pi \approx \tilde{\pi} = \mathrm{WProx}_{\beta^2 T\mathcal{H}, \beta^2/2}(\mathrm{WProx}^{-1}_{\beta^2 T\mathcal{H}, \beta^2/2}(\hat{\pi})), \tag{18}$$

where $\hat{\pi}$ is the empirical distribution defined by samples $\{X_i\}_{i=1}^N \sim \pi$, and $\tilde{\pi}$ is the generated distribution with an approximate score function. Here, the inverse WProx should be understood as a mapping, i.e., the set of densities ρ such that $\mathrm{WProx}_{\beta^2 T\mathcal{H}, \beta^2/2}(\rho) = \hat{\pi}$.

The connection between Wasserstein proximal operator and score-based generative modeling allows the use of kernel formulas to represent the score function. In particular, by appealing to kernel representation formula of the solution to the HJB equation (13), we can express the score function in terms of a similar kernel representation formula.

Proposition 3.1. *(Kernel representation formula for the score function). For initial condition $\eta(x,0) = \pi(x)$, the score function $\mathsf{s}(x,t) = \nabla \log \eta(x,t)$ in the denoising SDE* (2) *has the kernel representation formula*

$$\mathsf{s}(x,t) = -\nabla U(x,t) = \beta^2 \nabla \log \left((G_{\beta^2/2, T-t} * \pi)(x) \right) = \beta^2 \frac{(\nabla G_{\beta^2/2, T-t} * \pi)(x)}{(G_{\beta^2/2, T-t} * \pi)(x)} \tag{19}$$

$$\mathsf{s}(x,t) = -\beta^2 \frac{\int_{\mathbb{R}^d} \frac{x - x'}{\beta^2 (T-t)} G_{\beta^2/2, T-t}(x, x') \pi(x') \, dx'}{\int_{\mathbb{R}^d} G_{\beta^2/2, T-t}(x, x') \pi(x') \, dx'}. \tag{20}$$

Proof. The score function in (2) is the gradient of the logarithm of the solution to (17). The solution to (17) is related to the solution of (16) by a Cole-Hopf transform. The solution to (16) has a kernel representation formula as stated in Proposition 4 of [4] and (13). Taking the gradient of the kernel (13) gives us the representation formula for the score function. □

This kernel representation formula for the score function concisely describes the inductive bias of the score function. Proposition 3.1 implies that given parameters t and β, the Green's function of the associated PDE determines the kernel of the representation formula. Therefore, the problem at hand is to find a suitable approximation of the *true, unknown* density $\pi(x)$ to approximate the score function. In Section 4 we suggest two suitable approximations for the density function. Another kernel approximation to the score function using interacting particle systems was studied the context of approximating solution to the Fokker-Planck equation of the Langevin dynamics [36].

Remark 3.1. ***(Entropic regularization parameter defines equivalence classes)*** *For a given $h > 0$ in the Wasserstein proximal operator, there is a corresponding equivalence class of score-based models with diffusion coefficient β and time horizon T such that $\beta^2 T = h$. This perspective directly links the chosen parameters in SGMs to the step parameter in the Wasserstein proximal operator. The parameter $h = \beta^2 T$ can therefore be interpreted as a diffusion scaling.*

Remark 3.2. ***(Why is the Wasserstein proximal necessary?)*** *Recent state-of-the-art generative models are fundamentally described in terms of Wasserstein proximal formulations. For example, the optimal transport flow (OT-flow) [37] is the 2-Wasserstein proximal of the KL-divergence, and the (f, Γ) generative adversarial network is the 1-Wasserstein proximal of f-divergences [7]. By placing score-based and denoising diffusion generative models in terms of the 2-Wasserstein proximal, we make connections to other generative models and the power of proximal Wasserstein operators more broadly. In particular, recently the manifold learning property of proximal operators have been noted on* 1*-Wasserstein space [7, 8]. Placing SGMs in terms of the WPO may provide new explanations for its manifold learning properties [9, 10].*

Remark 3.3. ***(SGMs reverse the heat equation with MFG theory and initial data)*** *Given the heat equation $\partial_s \eta = \frac{\beta^2}{2}\Delta\eta$ and terminal condition $\eta(y, T)$, the task of solving for $\eta(y, s)$ for $s \in [0, T)$ is known as reversing the heat equation. Traditional discretization schemes are known to be numerically unstable as errors will accumulate and amplify [38]. Ostensibly, score-based generative models appear to provide a numerically stable method for reversing the heat equation; we, however, emphasize there is a subtle but significant difference between solving the backward heat equation and SGM. In contrast to reversing the heat equation, SGM also has* partial knowledge *of the initial distribution $\pi(y)$ in the form of samples. Having partial knowledge is empirically sufficient to make the problem better conditioned. This fact can be understood rigorously through MFG theory. The one-way coupled PDE systems* (16) *and* (17) *that arise as optimality conditions of the MFG formulation of SGMs (Theorem 7, [3]) and the equivalence of score-matching to the MFG system (Theorem 10, [3]) show that SGMs are well-posed. In particular, the Hamilton-Jacobi equation in* (16) *is a second order equation, and so standard regularity theory for viscous HJB equations imply that the PDE is well-posed [25]. Moreover, assuming regularity of the terminal condition, solutions to second order HJB equations are classical, as they are unable to form shocks (discontinuities). Therefore, the gradient of the HJB solution exists in a classical sense and the corresponding controlled Fokker-Planck that corresponds with the denoising SDE is well-posed too. MFG theory reveals that reversing the heat equation is a well-posed problem provided that the score function is known or can be learned.*

4 Deconstructing score-based generative models: a new WPO-informed kernel model

We have shown that score-based generative models are regularized Wasserstein proximal operators of cross-entropy. In this section we revisit the equivalent mean-field games formulation of score-based generative models, and use it to construct a new WPO-informed kernel model for the score function. Mean-field games in general are quite difficult to solve because of their nonlinear, coupled nature. For the Wasserstein proximal of cross-entropy, they are intriguingly decoupled, and reduce to a control problem. In particular, the Hamilton-Jacobi-Bellman equation together with its terminal condition,

$$\begin{cases} -\dfrac{\partial U}{\partial t} + \dfrac{1}{2}|\nabla U|^2 = \dfrac{\beta^2}{2}\Delta U \\ U(x,T) = -\log \pi(x), \end{cases} \tag{21}$$

is sufficient to characterize the solution of the mean-field game. Using the HJB equation, we will build a kernel-based model for the score function.

4.1 The empirical kernel model for the score function memorizes the training data

The equivalence of the HJB equation in (21) to a Fokker-Planck equation in (17) via a Cole-Hopf transformation was first used to construct kernel formulas for regularized Wasserstein operators of linear energies in [4]. From a probabilistic perspective, this is equivalent to the use of the probability transition kernel of the FP equation's corresponding stochastic process to compute the density function at future times. Given samples $\{Z_i\}_{i=1}^{N_{\text{train}}}$ from distribution π, the empirical distribution

$$\pi(\cdot) \approx \hat{\pi}(\cdot) = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \delta_{Z_i}(\cdot) \tag{22}$$

can serve as an approximation to the true distribution π. Appealing to the kernel representation formula for the solution to the HJB (13) is

$$\hat{U}(x,t) = -\beta^2 \log(G_{\beta^2/2,T-t} * \hat{\pi})(x) = -\beta^2 \log\left(\frac{1}{N_{\text{train}}}\sum_{i=1}^{N_{\text{train}}} G_{\beta^2/2,T-t}(x,Z_i)\right).$$

Note that this approximate solution only solves the HJB equation exactly for $t \in [0,T)$ as $\hat{U}(x,t)$ is *ill-defined* at $t = T$ due to the delta functions, so the terminal condition is not satisfied in the classical sense. The corresponding score function is still well-defined for $t \in [0,T)$ (i.e., not including $t = T$). Therefore, applying Proposition 3.1, we have the score formula

$$\mathsf{s}(x,t) = -\nabla\hat{U}(x,t) = \frac{(\nabla G_{\beta^2/2,T-t} * \hat{\pi})(x)}{(G_{\beta^2/2,T-t} * \hat{\pi})(x)} = -\beta^2 \frac{\sum_{i=1}^{N_{\text{train}}} \frac{x-Z_i}{\beta^2(T-t)} G_{\beta^2/2,T-t}(x,Z_i)}{\sum_{i=1}^{N_{\text{train}}} G_{\beta^2/2,T-t}(x,Z_i)}. \tag{23}$$

Using this empirical kernel formula directly, however, produces a score-based generative model that fails to generalize. In fact, when used directly as part of the reverse SDE, this model will *memorize* and *resample* from the training data $\{Z_i\}$. This behavior is the so-called memorization effect of score-based generative models and has been recently studied in both theoretical and applied contexts [4, 17, 18, 19]. In [9], it was shown that as long as the initial condition for the controlled Fokker-Planck equation $\rho_0(x)$ shared the same support as $\eta(y,T)$, and the empirical kernel formula is used as an approximation to the score function, then $\rho(x,T)$ will share the same support as $\hat{\pi}$. Moreover, it was shown in [32] that the denoising score-matching objective (5) is equivalent to explicitly score-matching to the empirical kernel formula. In particular, [9] noted that when neural nets are trained with the DSM objective (5), halting the training early is required to prevent the neural net from fitting exactly to the kernel formula. A sample complexity argument demonstrating this phenomenon was presented in [16] in which they showed that when a neural net is trained with the DSM objective function, the model fits exactly to the empirical kernel approximation for the score function (23). Appealing to the Wasserstein proximal notation in (18), using this empirical score score formula is equivalent applying the relation:

$$\hat{\pi} = \text{WProx}_{\beta^2 T\mathcal{H},\beta^2/2}(\text{WProx}^{-1}_{\beta^2 T\mathcal{H},\beta^2/2}(\hat{\pi})), \tag{24}$$

that is, using (23) simply returns the training data.

Moreover, in both the analysis and implementation of SGMs, early stopping of the denoising process, i.e. simulating (2) up to $t = T - \epsilon$ for some $\epsilon > 0$ is assumed to prove convergence of the method, or improved generalization [33, 10, 39, 16]. In continuous time, early stopping is equivalent to sampling from a mollified empirical distribution, i.e. $\eta(x, \epsilon) = (G_\epsilon * \hat{\pi})(x)$ or simply sampling from a Gaussian with variance $\epsilon^2\mathbf{I}$ around each training point. Early stopping alone may not be enough for effective generalization as ϵ is not always chosen in an informed manner. One can imagine simple examples where the extent of early stopping is applied is dependent on the data. In Section 5.3 we show numerical examples that show the impact of ϵ for generalization.

4.2 A kernel model that generalizes: resolving memorization

An accurate[2] score model should solve the PDE (21), in particular, including the terminal condition. In spite of the memorization effects of the kernel formula, they will be essential for producing a kernel-based SGM that generalizes. Crucially, the Cole-Hopf transformation of the kernel formula is closed under the evolution of the HJB equation. The kernel model, however, fails to match the terminal condition, which is why the empirical kernel model memorizes to the training data. Our approach is to simply alter the kernel formula so that it will be able to better match the terminal condition in (21) while still exactly solving the HJB equations.

One approach to matching the terminal condition is to mollify the kernel formula at time $t = T$ so that $\hat{\pi}$ is a density function with support that matches the true support of π rather than an empirical distribution. Convolving $\hat{\pi}$ with a Gaussian with a fixed bandwidth has the equivalent effect of early stopping with the empirical kernel score model (23). We will consider a more flexible mollification where the regularizing Gaussian has a state-dependent covariance matrix that is learned from the training data.

A better kernel model. Consider the following kernel-based approach that produces a better approximation to π in Proposition 3.1. Like the empirical distribution, Gaussian mixture model approximations to π are preferable in light of the representation formula (20) as the integral can be evaluated in closed form. We introduce a matrix-valued function to model the local precision matrix around each of the kernel centers. The inclusion of this precision matrix, in effect, yields a Gaussian mixture model, which is still closed under the evolution of the heat equation. Define a matrix-valued function $\mathbf{\Gamma}_\theta : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ to be a model for the precision matrix. Consider the model

$$\hat{\pi}_\theta(x; \{Z_i\}_{i=1}^N) = \frac{1}{N}\sum_{i=1}^N \frac{\det \mathbf{\Gamma}_\theta(Z_i)}{(2\pi)^{d/2}} \exp\left(-\frac{(x - Z_i)^\top \mathbf{\Gamma}_\theta(Z_i)(x - Z_i)}{2}\right), \tag{25}$$

where $N < N_{\text{train}}$. This formula is simply a generalization of the empirical distribution by Gaussian kernels where a subset of the training data is used as the kernel centers, and a local covariance matrix is learned around each center. In contrast to the empirical kernel formula that consists of a mixture of Dirac masses, this model is a *smooth* approximation which allows it to potentially generalize better. Moreover, as the HJB equation (21) is the fundamental characterization of SGMs, we show in Proposition 4.1 that the smooth WPO-informed kernel formula (25) is closed under the evolution of the HJB and, therefore, admits an analytical solution.

Proposition 4.1. *The HJB equation* (21) *with terminal condition* $\hat{U}(x, T) = -\beta^2 \log \hat{\pi}_\theta(x)$ *with* $\hat{\pi}_\theta(x)$ *given by* (25) *is solved by*

$$\hat{U}(x, t) = -\beta^2 \log \hat{\eta}_\theta(x, T - t) = -\beta^2 \log\left(\frac{1}{N}\sum_{i=1}^N \frac{\det \mathbf{\Gamma}_{T-t,\theta}(Z_i)}{(2\pi)^{d/2}} \exp\left(-\frac{(x - Z_i)^\top \mathbf{\Gamma}_{T-t,\theta}(Z_i)(x - Z_i)}{2}\right)\right), \tag{26}$$

$$\mathbf{\Gamma}_{t,\theta}(z) = (\mathbf{\Gamma}_\theta(z)^{-1} + \beta^2 t\mathbf{I})^{-1}.$$

[2] Meaning that the score model is able to express the inductive bias of score-based generative models, as defined by the Hamilton-Jacobi-Bellman equation (21).

Proof. It is easy to confirm that the terminal condition is satisfied exactly. Next, observe that

$$\begin{aligned}\partial_t \hat{U}(x,t) &= -\frac{\beta^2}{\hat{\eta}_\theta(x,T-t)}\frac{\partial}{\partial t}\hat{\eta}_\theta(x,T-t) = \frac{\beta^4}{\hat{\eta}_\theta(x,T-t)}\\ &\times \sum_{i=1}^{N} \frac{\det \boldsymbol{\Gamma}_{T-t,\theta}(Z_i)}{N(2\pi)^{d/2}} \exp\left(-\frac{(x-Z_i)^\top \boldsymbol{\Gamma}_{T-t,\theta}(Z_i)(x-Z_i)}{2}\right)\left(\frac{\|\boldsymbol{\Gamma}_{T-t,\theta}(Z_i)(x-Z_i)\|^2 - \beta^2 \mathrm{Tr}\boldsymbol{\Gamma}_{\theta,T-t}}{2}\right),\end{aligned}$$

$$\begin{aligned}\nabla \hat{U}(x,t) &= -\beta^2 \frac{\nabla \hat{\eta}_\theta(x,T-t)}{\hat{\eta}_\theta(x,T-t)}\\ &= \frac{\beta^2}{\hat{\eta}_\theta(x,T-t)} \sum_{i=1}^{N} \frac{\det \boldsymbol{\Gamma}_{T-t,\theta}(Z_i)}{N(2\pi)^{d/2}} \exp\left(-\frac{(x-Z_i)^\top \boldsymbol{\Gamma}_{T-t,\theta}(Z_i)(x-Z_i)}{2}\right)\left(\boldsymbol{\Gamma}_{T-t,\theta}(Z_i)(x-Z_i)\right),\end{aligned}$$

$$\frac{\beta^2}{2}\Delta \hat{U}(x,t) = -\frac{\beta^4}{2}\left(\frac{\Delta \hat{\eta}_\theta(x,T-t)}{\hat{\eta}_\theta(x,T-t)} - \left\|\frac{\nabla \hat{\eta}_\theta(x,T-t)}{\hat{\eta}_\theta(x,T-t)}\right\|^2\right),$$

where

$$\begin{aligned}&\Delta \hat{\eta}_\theta(x,T-t)\\ &= \sum_{i=1}^{N} \frac{\det \boldsymbol{\Gamma}_{T-t,\theta}(Z_i)}{N(2\pi)^{d/2}} \exp\left(-\frac{(x-Z_i)^\top \boldsymbol{\Gamma}_{T-t,\theta}(Z_i)(x-Z_i)}{2}\right)\left(\|\boldsymbol{\Gamma}_{T-t,\theta}(Z_i)(x-Z_i)\|^2 - \beta^2 \mathrm{Tr}\boldsymbol{\Gamma}_{\theta,T-t}\right).\end{aligned}$$

To differentiate the determinant of a symmetric matrix-valued function with respect to a parameter t, we used the fact that $\mathbf{A}(t)$ for $t \in \mathbb{R}$, $\partial_t \log\det \mathbf{A}(t) = \mathrm{Tr}\left(\mathbf{A}(t)^{-1}\partial_t \mathbf{A}(t)\right)$ [40]. Matching terms, we see that $\frac{\beta^2}{2}\Delta \hat{U}(x,t) = -\partial_t \hat{U}(x,t) + \frac{1}{2}\|\nabla \hat{U}(x,t)\|^2$. □

The WPO-informed kernel model presented in Proposition 4.1 inherits the inductive bias of the score function as described by the kernel representation formula in Proposition 3.1. See Figure 1 for a graphical explanation of the relationship between the model in (26) and (20). The form of the WPO-informed kernel model (25) and the result of Proposition 4.1 further supports the point that the model provides an *explainable* characterization of SGMs. As noted in [3] and [41], SGMs inherently solve the HJB equation (21). Proposition 4.1 shows that *given* the terminal condition is a kernel formula (Gaussian mixture model), the score function is uniquely determined by the gradient of the solution (26). Therefore, there is no need to perform score-matching for all $t \in [0,T]$, as it is done in all the score-matching objectives (3),(4),(5) — enforcing the terminal condition is sufficient to determine the score function $\hat{s}(x,t)$ for all $t \in [0,T]$. In the next section, we show that performing score-matching only for the terminal time $t = T$ in (25) is sufficient. Moreover, the implicit score-matching objective at the terminal time is used to train the local precision matrix model. The form of the model explicitly encodes the manifold learning properties of score-based generative models. In particular, we highlight Remark 4.1 which provides a connection between the precision matrix in the WPO-informed model (25) and Riemannian manifolds.

Remark 4.1. ***(Learning the local precision matrices is manifold learning)*** *Incorporating a pre-processing step that learns the underlying data manifold, i.e. the latent space, has been empirically found to improve the generative qualities of SGMs [27, 28]. Moreover [9] observed and proved that SGMs inherently learn data manifolds. We explicitly exploit the results of [9] by building a new kernel-based model* (29) *that learns information about the data manifold through local precision matrices. Manifolds embedded in Euclidean space are described by Riemannian metric tensors, which are a family of positive semidefinite symmetric matrices [26]. These metrics correspond to the learned precision matrices. The resulting density function is defined with respect to the Lebesgue measure of the data manifold.*

Enforcing the terminal condition with implicit score-matching efficiently learns the data manifold. Our kernel-based model is constructed to explicitly solve the HJB for $t \in [0,T)$; the remaining task is to learn the local precision matrices around the kernel centers so that the terminal condition is satisfied. To this end, we use the terminal condition to construct a loss function. Imposing some L^2 loss between the approximate $\hat{U}_\theta(x,T)$ and true $U(x,T)$ requires knowledge of the density and normalizing constant of π, which we do not have. Remarkably, however, the simplest way to enforce the terminal condition is to match the *gradient* of the terminal condition, which is equivalent to implicit score-matching (4) only at the terminal time $t = T$. Score-matching is not needed for $t < T$ since the kernel solution to the HJB equation will optimize the ISM objective function exactly because of the mean-field game formulation. From Theorem 10 in [3], it is shown that ISM is equivalent to the SGM MFG optimality condition (16). Moreover, Proposition 4.1 shows that the kernel-based model solves the HJB exactly.

Consider the optimization problem

$$\min_\theta \int_{\mathbb{R}^d} |\nabla U(x,T) - \nabla \hat{U}_\theta(x,T)|^2 \pi(x)\,\mathrm{d}x \implies \min_\theta \int_{\mathbb{R}^d} \left(|\nabla \log \pi_\theta(x)|^2 + 2\Delta \log \pi_\theta(x)\right) \pi(x)\,\mathrm{d}x. \tag{27}$$

Solving this optimization problem has the effect of learning the local covariance matrix, which has the same effect as learning the manifold on which the data distribution is supported. The empirical kernel formula (22) is not well-defined as a function at the terminal time. Moreover, the derivative of the empirical kernel formula for $t = T$ is not defined, which implies that derivative cannot be taken and the integration by parts formula does not apply. There is, however, an implicit regularity assumption imposed on the target distribution when constructing SGMs since the integration by parts formula is used to derive the ISM (4). The implicit regularity assumption is critical for the resulting generative model to generalize.

Let $y(\,\cdot\,;\theta) : \mathbb{R}^d \to \mathbb{R}^N$ be a vector-valued function where the i-th component is

$$y_i(x;\theta) = -(x - Z_i)^\top \mathbf{\Gamma}_\theta(Z_i)(x - Z_i) + \log\det \mathbf{\Gamma}_\theta(Z_i) - \frac{d}{2}\log 2\pi - \log N, \tag{28}$$

so that the density is $\hat{\pi}_\theta(x) = \sum_{i=1}^N \exp(y_i(x;\theta))$, and the score function is

$$\hat{\mathsf{s}}_\theta(x) = \nabla \log \hat{\pi}_\theta(x) = \frac{\sum_{j=1}^N \nabla y_j(x;\theta)\exp(y_j(x;\theta))}{\sum_{j=1}^N \exp(y_j(x;\theta))} = -\sum_{j=1}^N \mathbf{\Gamma}_\theta(Z_j)\,(x - Z_j)\,\sigma(y(x;\theta))_j, \tag{29}$$

where $\sigma : \mathbb{R}^N \to \mathbb{R}^N$ is the softmax function, $\sigma(y)_i = \frac{\exp(y_i)}{\sum_{j=1}^N \exp(y_j)}$. Matching the gradient of the terminal condition (27) and using the fact that $\Delta \log \pi_\theta = \pi_\theta^{-1}\Delta\pi_\theta - |\nabla \log \pi_\theta|^2$ yields the optimization problem

$$\min_\theta \int_{\mathbb{R}^d} \left(2\pi_\theta^{-1}\Delta\pi_\theta - |\nabla \log \pi_\theta(x)|^2\right) \pi(x)\,\mathrm{d}x, \tag{30}$$

where

$$\pi_\theta(x)^{-1}\Delta\pi_\theta(x) = \sum_{i=1}^N \left(|\nabla y_i(x;\theta)|^2 + \Delta y_i(x;\theta)\right)\sigma(y(x;\theta))_i, \tag{31}$$

$$\nabla y_i(x;\theta) = -\mathbf{\Gamma}_\theta(Z_j)(x - Z_j), \quad \Delta y_i(x;\theta) = -\mathrm{Tr}(\mathbf{\Gamma}_\theta(Z_j)).$$

A neural net is used to model $\mathbf{\Gamma}_\theta(z)$ via the implicit score matching procedure at the terminal condition (30). This produces the WPO-informed kernel model for the score function (25) that, if used in conjunction with Proposition 4.1 and the denoising SDE (2), will exactly produce samples from the kernel formula (25). We note, however, in Remark 4.3 that since (25) is a Gaussian mixture model, no simulation of SDEs is needed and samples can be produced directly from the WPO-informed kernel model. What allows the WPO-informed kernel formula to generalize and avoid memorization is its smoothness and regularity at the terminal condition. Moreover, we show that judicious use of a neural net can improve training — rather than replacing the entire score function with a neural net, we preserve mathematical structure informed by the MFG PDEs associated with SGMs by embracing the kernel formula, and introduce a neural net to model the local precision matrices only. We demonstrate these claims via numerical examples in Section 5. Moreover, in Section 6.2 we show that PDE and kernel structure for the score function may yield informed neural net architecture for scalable implementations of the WPO-informed kernel model.

Remark 4.2. ***(Implicit score-matching without autodifferentiation)*** *In contrast to previous deep learning implementations of implicit score-matching [31], which requires autodifferentiation packages and the Hutchinson estimator to compute the divergence of the score function, the WPO-kernel formula admits exact formulas for gradient, divergences, and Laplacians of the kernel formula and its Cole-Hopf transform.*

Remark 4.3. ***(No simulation of SDEs required)*** *The kernel-based model* (25) *obviates the need for simulating SDEs since the mixture model can be sampled directly. One, however, can still derive a score model for the denoising SDE with standard choices of the noising process (similar to Proposition 4.1). The noising process is typically chosen to be a linear SDE, so the transition kernel is Gaussian, and the score function can be found by convolving the transition kernel with* $\hat{\pi}_\theta(x)$.

Parametrizing the precision matrix. We outline one (but not the only) approach for parameterizing the precision matrix [42] which we apply in our numerical examples in Section 5. We parametrize the precision matrix $\mathbf{\Gamma}_\theta \in \mathbb{R}^{d\times d}$ in terms of the Cholesky factors: $\mathbf{\Gamma}_\theta(x) = \mathbf{L}_\theta(x)\mathbf{L}_\theta(x)^\top$, where $\mathbf{L}_\theta$ is a lower triangular matrix. The entries of the Cholesky factor is populated by the outputs of a feedforward neural network $\psi_\theta : \mathbb{R}^d \to \mathbb{R}^{d(d+1)/2}$.

We summarize the algorithm in Algorithm 1. The kernel model can be directly sampled just as a Gaussian mixture model would. This algorithm is an explainable reformulation of score-based generative modeling, derived by explicitly incorporating the inherent kernel structure. The model clarifies the manifold learning properties of SGM, the use of early stopping, and obviates the need for simulating the denoising SDE.

Algorithm 1 Learning WPO-informed kernel models

```
Input: Samples $\{Z_i\} \sim \pi$, Cholesky factor entries neural network $\psi_\theta : \mathbb{R}^d \to \mathbb{R}^{d(d+1)/2}$
Define precision matrix $\mathbf{\Gamma}_\theta(x) = \mathbf{L}_\theta(x)\mathbf{L}_\theta(x)^\top$
Set $\hat{\pi}_\theta \leftarrow$ (25),$\nabla \log \hat{\pi}_\theta \leftarrow$ (29), $\hat{\pi}_\theta^{-1}\Delta\hat{\pi}_\theta \leftarrow$ (31) (Requires no autodifferentiation)
Find the optimal precision matrix $\theta^* \leftarrow \arg\min_\theta \frac{1}{N}\sum_{i=1}^{N} 2\hat{\pi}_\theta(Z_i)^{-1}\Delta\hat{\pi}_\theta(Z_i) - |\nabla \log \hat{\pi}_\theta(Z_i)|^2$
return $\hat{\pi}_{\theta^*}(x)$
```

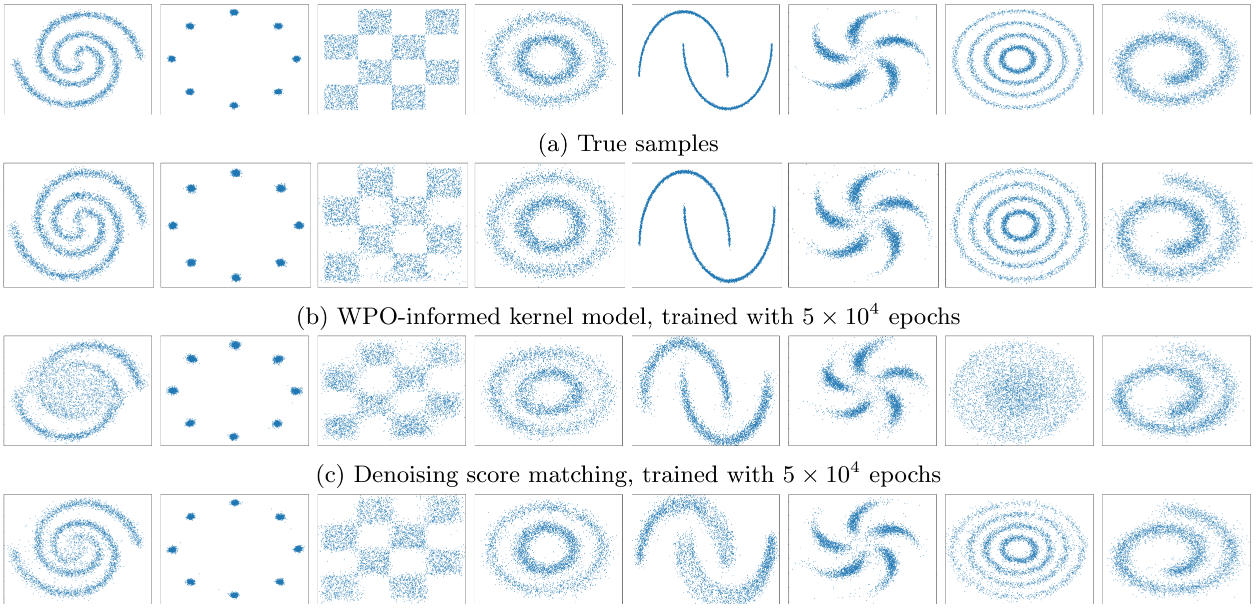
Remark 4.4. *(Early stopping revisited) Early stopping is a frequently used strategy to improve the sample quality of score-based generative models [16]. When simulating the denoising SDE, each trajectory is not simulated for the full time interval and is stopped early. This has the effect of smoothing the generative distribution and has been observed to aid in generalization. The precise amount of early stopping requires tuning. We may interpret our kernel-based formula as a generalization of early stopping, in which the local precision matrices are learned from data. It can also be interpreted as a way to choose the optimal early stopping parameter. Consider a simplified model where the local covariance matrix is identical over all space and is simply* $h\mathbf{I}$*. Then the optimal covariance solves* (30) *where* $\theta^* = h^*$ *with the score model* (29) *where* $\mathbf{\Gamma} = {h^*}^{-1}\mathbf{I}$*. Sampling from this tuned distribution is the same as early stopping at* $t = h$.

5 Numerical examples

We conduct illustrative numerical experiments on synthetic datasets to demonstrate the effectiveness of our WPO-informed kernel model. We emphasize that the method here is not optimized to be scalable— rather we implement the kernel model to demonstrate its manifold learning and generalization properties. In particular, we show that the WPO kernel model trains faster and intrinsically provides a density estimate. We also illustrate the explicit manifold learning properties of the kernel formula and demonstrate its superiority to early stopping.

5.1 WPO-informed kernel model trains faster and provides density estimation

We implement the WPO-informed kernel model according to Algorithm 1, where the function ψ_θ that models the entries of the Cholesky factor comprise is a feedforward neural network with five hidden layers of 64 nodes and a GeLU activation function. The terminal score-matching optimization problem is solved via



(a) True samples

(b) WPO-informed kernel model, trained with 5×10^4 epochs

(c) Denoising score matching, trained with 5×10^4 epochs

(d) Denoising score matching, trained with 10^6 epochs

Figure 2: 2500 samples generated via different models when training dataset size is limited to 5×10^4.

stochastic gradient descent with batch size 64 and training dataset of size $N_{\text{train}} = 5 \times 10^4$. The kernel centers $\{Z_i\}_{i=1}^N$ where $N = 5000$ are chosen randomly from the training data. For comparison, we also train a standard SGM (2) with the denoising score-matching objective (5) to train the score neural network. The neural network for the score function $\mathsf{s}_\theta(x, t)$ has the same hidden architecture as ψ_θ and is trained with the same 5×10^4 training samples.

In Figure 2, we show the generated samples from the two models and compare it with samples from the true data distribution. Observe in Figure 2b that the WPO-informed kernel model is able to nearly exactly reproduce the true distribution with just 5×10^4 steps of SGD. In contrast, in the Figure 2c, we show that the SGM learned via denoising score matching produces poor quality samples in 5×10^4 steps, while Figure 2d shows that it can reproduce the samples after 10^6 training steps. Notice, however, that the two thin moon dataset is poorly approximated by denoising score matching SGM even after 10^6 training steps. This is likely due to the fact that the neural net used to approximate the score function is unable to capture the irregular score function that arises from the thin moons. In contrast, the WPO-informed kernel model easily models the low-dimensional nature of the thin moons.

Furthermore, in contrast to standard score-based generative models, the WPO-informed kernel model inherently provides a density estimate. In Figure 3 we plot the kernel density approximation that corresponds with the generated samples Figure 4. We emphasize that these density plots are *not* reconstructed densities from the generated samples, rather they are directly provided by the WPO-informed kernel model in (25).

To demonstrate that the WPO-informed kernel model is at least scalable to moderate dimensions even with the crude parametrization of the Cholesky factors of the precision matrices, we apply our method to the 3D swissroll dataset noisily embedded in a six-dimensional space. In Figure 4 we demonstrate the WPO-informed kernel model is able to produce a density estimate of the data distribution.

5.2 Learning the local covariance learns the data manifold

To highlight the manifold learning properties of the WPO-informed kernel model, in Figure 5 we plot the density estimate of the two thin moons example along with plots of evaluations of the learned local covariance (inverse precision) matrices. We plot the eigenvectors of each covariance matrix scaled by its corresponding eigenvalue. Notice that the orientation of the ellipses change with location and that the axes of the ellipse are not identical. The precision matrix that is learned is, in effect, learning the Riemannian metric of the

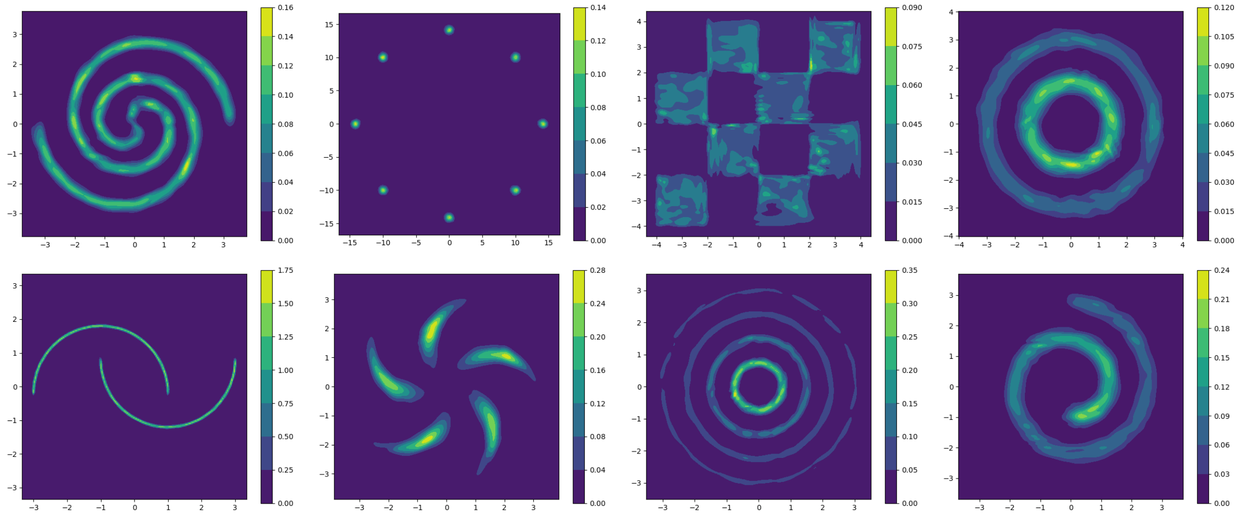
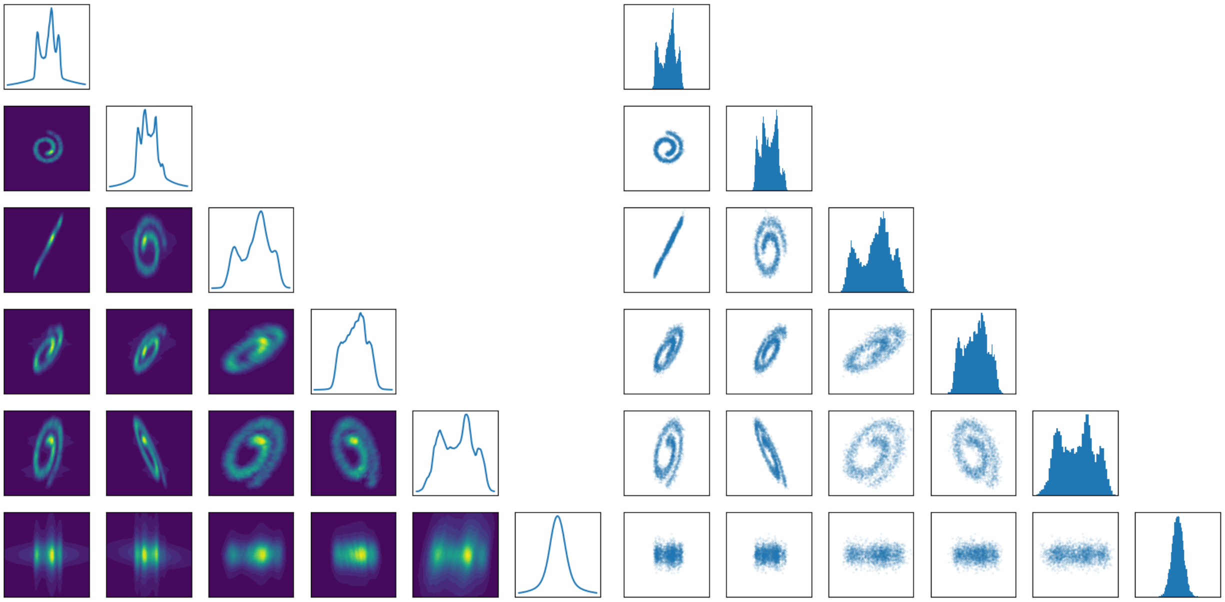


Figure 3: Density plots constructed by evaluating the kernel density estimated using WPO-informed kernel model as in experiments shows in Figure 2. These density plots are not reconstructed densities from samples.



(a) WPO-informed kernel model 2D marginal densities.

(b) True samples.

Figure 4: Six dimensional example: 3D swissroll noisily embedded in a 6D space. Proof of concept that the WPO-informed kernel model (25) is at least scalable to moderate dimensions.
We train the WPO-informed kernel model with 5000 kernel centers with 10^6 training points. A Cholesky factor model is a feedforward neural net with 6 hidden layers and 64 nodes per layer. The model is trained via stochastic gradient descent with batch size 64 over 10^5 iterations. The 2D marginal densities are not reconstructed from data.

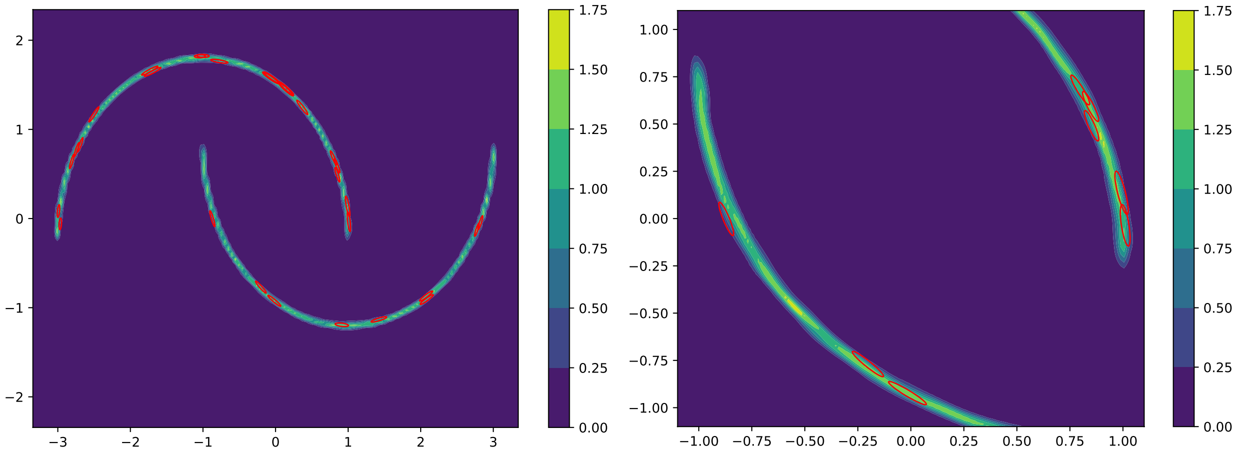


Figure 5: We use ellipses to represent the local covariance matrices obtained from the WPO-informed kernel model, which shows our trained model reveals the underlying data manifold. The model is trained with the two moons dataset, as the experiment depicted in Figure 3. We draw 25 samples $Z_i, i = 1, ..., 25$ from the trained kernel centers, accessing the learned covariance matrices by evaluating the trained Cholesky factorization $\mathbf{\Gamma}_\theta(x)^{-1}$ at $x = Z_i$ for $i = 1, ..., 25$. Each ellipse is centered at Z_i, and its orientation and axes are determined by the eigenvectors and eigenvalues of the corresponding covariance matrix. On the right, we present a zoomed-in plot of the left figure.

data manifold [26]. Moreover, as the data distribution truly becomes lower dimensional, i.e., the two moons become simply two lines, then the length of the shorter axis of each ellipse will tend towards zero.

5.3 Manifold learning generalizes better than early stopping and prevents memorization

We further emphasize that the proposed WPO-informed kernel model generalizes better than the empirical kernel formula (23) due to the WPO-informed model's manifold learning capabilities. From [16, 9], it is known that a neural net trained with the denoising score matching objective (5) will, as the model complexity and training dataset increases, converge to the kernel score formula in (23), which is known to produce a generative model that memorizes the training data and will not generalize. However, in practical implementation, early stopping of the denoising diffusion process, i.e., simulating (2) for $t \in [0, T - \epsilon]$ for some $0 < \epsilon \ll 1$ is often performed so that the SGM generalizes. As discussed in Section 4.1 and in [16], in the continuous time, early stopping has the effect of sampling from $(G_{\beta^2/2,\epsilon} * \hat{\pi})(x)$.

In Figure 6 we perform an illustrative experiment showing the superior generalization capabilities of the WPO-informed kernel model. Sampling from $(G_{\beta^2/2,\epsilon} * \hat{\pi})(x)$ is effectively sampling from the original training dataset added to an isotropic Gaussian noise. When ϵ is small, the generative distribution is quite rough and has patchy support. When ϵ is large, this has the effect of sampling from a very smoothed data distribution, which means the model lacks the expressiveness to approximate the true data distribution. In contrast, we plot the result of the WPO-informed kernel model and show that it is able to adaptively learn the local precision matrices and, therefore, better approximate the data distribution and generalize better.

6 Discussion: Resolving memorization and scalable implementations of the WPO-informed kernel model

We presented a fundamental characterization of score-based generative models with SDEs in terms of the **Wasserstein proximal operator (WPO) of cross-entropy**. We showed that, through **mean-field games** and the **Cole-Hopf transformation**, the entropically regularized WPO precisely yields the canonical formulation of SGMs where the noising process is a Brownian motion. The optimality conditions of

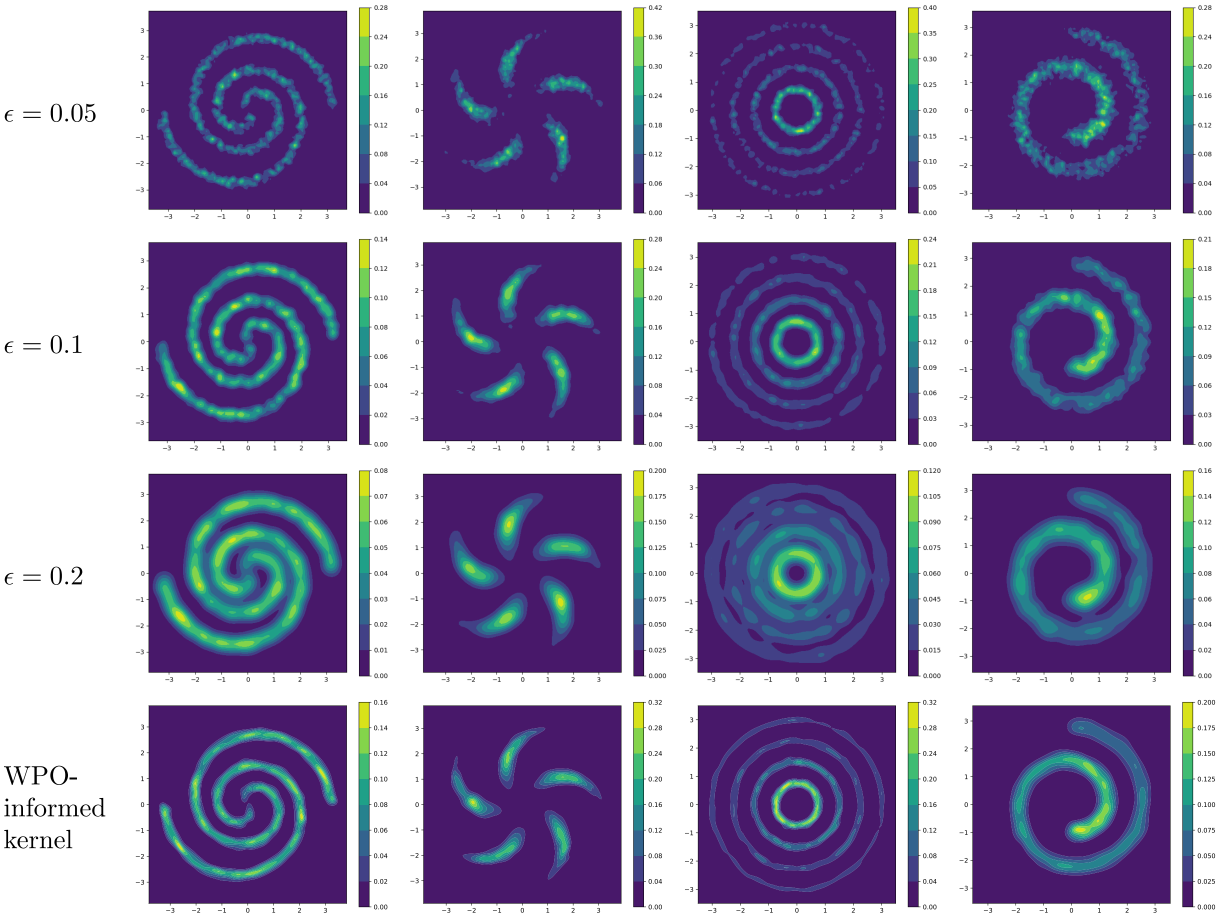


Figure 6: Comparison of density estimates between the empirical kernel model and WPO-informed kernels. All figures are estimated using a set of 2500 samples for four datasets. In the first three rows, we employ an identity covariance with a constant scalar ϵ. In the last row, we utilize the WPO-informed kernel model trained on 5000 sample centers over 10^5 epochs. Subsequently, plot the density using the trained model.

the mean-field game yield an **Hamilton-Jacobi-Bellman equation** that can be exactly expressed with a kernel representation formula (20) (Proposition 3.1). Given finite training data, application of the kernel formula using the empirical distribution (23) for SGM will memorize the training data. We proposed a WPO-informed kernel model that is based on learning local precision matrices around some subset of the training data, which has the effect of **learning the manifold** of the underlying data distribution. The local precision matrices, and therefore the data manifold, are learned by fitting to the **terminal condition** of the HJB via implicit score-matching. We demonstrated through numerical examples that our approach results in a score model that is *interpretable* and **learns faster with less data**. Here we summarize how the WPO-informed kernel model resolves memorization and how it may be scaled to higher dimensions via a deep learning connection.

6.1 Resolving memorization in score-based generative models

The memorization phenomenon of score-based generative models has been observed in simple toy models [9, 16], in small image datasets [19], large image datasets [17], and in practical text-to-image applications [18]. Moreover, [9] studies the memorization phenomenon through SGM's manifold detecting behavior, showing that under suitable conditions, the optimal function that minimizes the DSM objective will result in a generative model that matches the support of the training data. In [16] the authors show that a neural net with sufficient complexity will, with enough training time, learn the empirical score kernel formula (23) very well. The learned score function yields a generative model whose generalization properties amount to early stopping of the denoising diffusion process, and that the produced samples come from a kernel density estimate. The score formula coming from the empirical distribution is typically cast in a negative light due its memorization effects. Our WPO-informed approach to SGM *embraces* the kernel formula as kernels encode the core mathematical structure of score-based generative models, and we show that the kernel formula can be modified to avoid memorization.

We summarize our contribution in explaining and resolving memorization effects in score-based generative modeling. The key reason for why SGMs memorize the training data is demonstrated by the optimal score function for the denoising score matching optimization problem (5) which, assuming a finite sample, can be written in closed form using a kernel formula (23) based on the empirical distribution on the training data (22). When the denoising diffusion process (2) is simulated with the empirical score kernel formula (23), the system will sample only from the the empirical distribution, thereby exhibiting memorization of the training data.

Our WPO-informed kernel model for the score function relies on the connection between score-based generative models and the Wasserstein proximal operator (15) via a mean-field game (16). The form of the resulting MFG partial differential equations informs the Cole-Hopf and kernel structures of its solutions, yielding Proposition 3.1, which provides a general kernel representation formula (20) the score function. One approximation for (20) is via the empirical distribution of the training samples (23), but is not the only choice. The WPO-informed kernel formula (25) provides a *smooth* approximation to the data distribution where local precision matrices around certain kernel centers are learned. The smoothness of the approximation is key to avoid memorization and improving generalization. The precision matrices are learned by only performing implicit score matching *at the terminal time.* This is not possible with the denoising score matching objective as it will simply fit to the training data. The use of the implicit score matching objective is only possible with our model because it is *smooth*, and by the fact in Proposition 4.1, shows that it is *sufficient* to only learn the score at the terminal time as the kernel formula provides a closed-form solution for the score at all future times. In fact, we welcome a description of the WPO-informed kernel model as simply *a Gaussian mixture model that is trained using the implicit score matching objective.*

6.2 Scaling the WPO-informed kernel model for high dimensional applications: a bespoke neural network architecture

While the kernel-based score model yields an interpretable model that is constructed to respect its fundamental mathematical structure, they are generally not scalable due to the computational cost of evaluating the kernels. Here, we study the kernel-based model (25) and show that it admits a neural network interpretation, thereby providing means for scaling the kernel-based model to high-dimensional applications. In other words, we provide preliminary explorations of what are suitable neural network approximations to the kernel representation formula in Proposition 3.1. The resulting neural network architecture not only inherits the kernel structure from the WPO-informed model but, more fundamentally, the PDE structure informed by the Cole-Hopf formula. The resulting bespoke neural network then becomes yet another way to approximate the kernel representation formula in Proposition 3.1.

For $x \in \mathbb{R}^d$, let $(x \otimes x)$ be the column-wise Kronecker product

$$(x \otimes x) = \begin{bmatrix} x_1^2 & x_1x_2 & \cdots & x_1x_d & x_2^2 & \cdots & x_2x_d & \cdots & \cdots & x_d^2 \end{bmatrix}^\top \in \mathbb{R}^{d^2}. \tag{32}$$

Define a lifting operator $\mathcal{T} : \mathbb{R}^d \to \mathbb{R}^{d^2+d}$, $\mathcal{T}(x) = \tilde{x} = \left[(x \otimes x)^\top, \, x^\top\right]^\top$. Recall the kernel score model (29) and notice that it may be written in terms of the softmax function, which implies a neural network connection. Indeed, observe that we may rewrite the exponent in (29) as a linear function in the lifted space, namely

$$\mathbf{y}_\theta(x) = \mathbf{A}_\theta \mathcal{T}(x) + \mathbf{b}_\theta \, ,$$

where $\mathbf{A}_\theta = \begin{bmatrix}\mathbf{G}_\theta & \mathbf{H}_\theta\end{bmatrix}^\top \in \mathbb{R}^{N\times(d^2+d)}$, with matrices $\mathbf{G}_\theta \in \mathbb{R}^{N\times d^2}$, $\mathbf{H}_\theta \in \mathbb{R}^{N\times d}$, $\mathbf{b}_\theta \in \mathbb{R}^N$, and

$$\begin{aligned}
\mathbf{G}_\theta &= \begin{bmatrix}\mathrm{vec}(\boldsymbol{\Gamma}_\theta(Z_1)) & \mathrm{vec}(\boldsymbol{\Gamma}_\theta(Z_2)) & \cdots & \mathrm{vec}(\boldsymbol{\Gamma}_\theta(Z_N))\end{bmatrix}^\top \\
\mathbf{H}_\theta &= \begin{bmatrix}\boldsymbol{\Gamma}_\theta(Z_1)Z_1 & \boldsymbol{\Gamma}_\theta(Z_2)Z_2 & \cdots & \boldsymbol{\Gamma}_\theta(Z_N)Z_N\end{bmatrix}^\top \\
(\mathbf{b}_\theta)_i &= \log\det\boldsymbol{\Gamma}_\theta(Z_i) - \frac{d}{2}\log 2\pi - \log N.
\end{aligned} \tag{33}$$

Here, vec : $\mathbb{R}^{N\times d} \to \mathbb{R}^{Nd}$ is the vectorization operator that concatenates columns of a matrix into one vector. The kernel score formula (29) is then written as

$$\begin{aligned}
\mathsf{s}_\theta(x) &= (\nabla\mathbf{y}_\theta(x))\sigma\left(\mathbf{A}_\theta\mathcal{T}(x) + \mathbf{b}_\theta\right), \\
\nabla\mathbf{y}_\theta(x) &= \mathbf{F}_\theta x - \mathbf{H}_\theta^\top, \\
\mathbf{F}_\theta(x) &= \begin{bmatrix}\boldsymbol{\Gamma}_\theta(Z_1)x & \boldsymbol{\Gamma}_\theta(Z_2)x & \cdots & \boldsymbol{\Gamma}_\theta(Z_N)x\end{bmatrix}.
\end{aligned} \tag{34}$$

We see that the score kernel model can be interpreted in terms of a shallow neural network where the inputs are elements of the lifted space $\mathbb{R}^{(d^2+d)N}$. This is a specialized, *bespoke* neural network architecture derived from the WPO-informed kernel formula (25) that we contend may learn score models in a scalable fashion in high-dimensional applications. For example, typical deep learning implementations of implicit score-matching requires the use of autodifferentiation packages to compute or approximate the divergence of the score model [31]. The specialized kernel formula will admit closed form expressions for the divergence via its connection to the kernel formulas noted in Remark 4.2. Moreover, the score functions in the family of neural networks with the specialized architecture (34) at any time $t \in [0, T)$ can be derived via Proposition 4.1.

We emphasize that there is a direct correspondence between learning the parameters of the neural network, i.e., $\mathbf{A}_\theta, \mathbf{b}_\theta$ to learning the local precision matrices, $\boldsymbol{\Gamma}(Z_i)$, in the kernel density formula. In particular, notice that entries of $\mathbf{A}_\theta$ require knowledge of the local precision matrices around kernel centers as well as the centers Z_i themselves, while learning $\mathbf{b}_\theta$ corresponds with a normalizing factor depending on the local precision matrices. The neural network interpretation suggests that the kernel centers can be *learned* rather than placed *a priori* as we did in Sections 4 and 5. These properties give further intuitive justification for how SGMs learn manifolds of the underlying data distribution — when a generic neural net learns the score function, it is learning (1) the implicit mathematical structure that is inherent in the score function, i.e., the lifting transform $\mathcal{T}$, (2) the kernel centers, and (3) the local precision matrix around each kernel center. Furthermore, in contrast with typical neural network training procedures, we know the explicit relationships within parameters $\mathbf{A}_\theta$ and $\mathbf{b}_\theta$, meaning that their relation may be imposed within the training process and may accelerate their learning. In particular, the learning of the lifting operator $\mathcal{T}$ is, in effect, learning an explicit mathematical relationship that is invariant with respect to the training data. Learning this explicit relationship would introduce additional statistical errors than if the relation were included in the model *a priori* to training.

References

[1] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[2] S. Osher, H. Heaton, and S. Wu Fung, "A Hamilton–Jacobi-based proximal operator," *Proceedings of the National Academy of Sciences*, vol. 120, no. 14, p. e2220469120, 2023.

[3] B. J. Zhang and M. A. Katsoulakis, "A mean-field games laboratory for generative modeling," *arXiv preprint arXiv:2304.13534*, 2023.

[4] W. Li, S. Liu, and S. Osher, "A kernel formula for regularized Wasserstein proximal operators," *Research in Mathematical Sciences*, vol. 10, p. 43, 2023.

[5] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *arXiv preprint arXiv:2011.13456*, 2020.

[6] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[7] J. Birrell, P. Dupuis, M. A. Katsoulakis, Y. Pantazis, and L. Rey-Bellet, "(f-Γ)-divergences: Interpolating between f-divergences and integral probability metrics," *Journal of Machine Learning Research*, vol. 23, no. 39, pp. 1–70, 2022.

[8] H. Gu, P. Birmpa, Y. Pantazis, L. Rey-Bellet, and M. A. Katsoulakis, "Lipschitz-regularized gradient flows and generative particle algorithms for high-dimensional scarce data," 2023.

[9] J. Pidstrigach, "Score-based generative models detect manifolds," *Advances in Neural Information Processing Systems*, vol. 35, pp. 35852–35865, 2022.

[10] V. De Bortoli, "Convergence of denoising diffusion models under the manifold hypothesis," *arXiv preprint arXiv:2208.05314*, 2022.

[11] D. Kwon, Y. Fan, and K. Lee, "Score-based generative modeling secretly minimizes the Wasserstein distance," *Advances in Neural Information Processing Systems*, vol. 35, pp. 20205–20217, 2022.

[12] T. Dockhorn, A. Vahdat, and K. Kreis, "Score-based generative modeling with critically-damped langevin diffusion," *arXiv preprint arXiv:2112.07068*, 2021.

[13] G. Franzese, S. Rossi, L. Yang, A. Finamore, D. Rossi, M. Filippone, and P. Michiardi, "How much is enough? a study on diffusion times in score-based generative models," *Entropy*, vol. 25, no. 4, p. 633, 2023.

[14] Q. Zhang and Y. Chen, "Fast sampling of diffusion models with exponential integrator," *arXiv preprint arXiv:2204.13902*, 2022.

[15] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," *arXiv preprint arXiv:2303.01469*, 2023.

[16] S. Li, S. Chen, and Q. Li, "A good score does not lead to a good generative model," *arXiv preprint arXiv:2401.04856*, 2024.

[17] G. Somepalli, V. Singla, M. Goldblum, J. Geiping, and T. Goldstein, "Diffusion art or digital forgery? investigating data replication in diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6048–6058, 2023.

[18] G. Somepalli, V. Singla, M. Goldblum, J. Geiping, and T. Goldstein, "Understanding and mitigating copying in diffusion models," *arXiv preprint arXiv:2305.20086*, 2023.

[19] X. Gu, C. Du, T. Pang, C. Li, M. Lin, and Y. Wang, "On memorization in diffusion models," *arXiv preprint arXiv:2310.02664*, 2023.

[20] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramer, B. Balle, D. Ippolito, and E. Wallace, "Extracting training data from diffusion models," in *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253–5270, 2023.

[21] S. A. Thompson, "We asked A.I. to create the Joker. It generated a copyrighted image," *The New York Times*, January 2024. Accessed: February 9, 2024.

[22] S. A. Thompson, "The Times sues OpenAI and Microsoft over A.I. use of copyrighted work," *The New York Times*, December 2023. Accessed: February 9, 2024.

[23] A. Salim, A. Korba, and G. Luise, "The wasserstein proximal gradient algorithm," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12356–12366, 2020.

[24] A. T. Lin, W. Li, S. Osher, and G. Montúfar, "Wasserstein proximal of GANs," in *International Conference on Geometric Science of Information*, pp. 524–533, Springer, 2021.

[25] L. C. Evans, *Partial differential equations*, vol. 19. American Mathematical Society, 2022.

[26] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[27] A. Vahdat, K. Kreis, and J. Kautz, “Score-based generative modeling in latent space,” in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 11287–11302, Curran Associates, Inc., 2021.

[28] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” *CoRR*, vol. abs/2112.10752, 2021.

[29] F. Santambrogio, “Optimal transport for applied mathematicians,” *Birkäuser, NY*, vol. 55, no. 58-63, p. 94, 2015.

[30] B. D. Anderson, “Reverse-time diffusion equation models,” *Stochastic Processes and their Applications*, vol. 12, no. 3, pp. 313–326, 1982.

[31] Y. Song, S. Garg, J. Shi, and S. Ermon, “Sliced score matching: A scalable approach to density and score estimation,” in *Uncertainty in Artificial Intelligence*, pp. 574–584, PMLR, 2020.

[32] P. Vincent, “A connection between score matching and denoising autoencoders,” *Neural computation*, vol. 23, no. 7, pp. 1661–1674, 2011.

[33] Y. Chen, T. T. Georgiou, and M. Pavon, “On the relation between optimal transport and Schrödinger bridges: A stochastic control viewpoint,” *Journal of Optimization Theory and Applications*, vol. 169, pp. 671–691, 2016.

[34] T. M. Cover, *Elements of information theory.* John Wiley & Sons, 1999.

[35] R. Y. Rubinstein and D. P. Kroese, *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*, vol. 133. Springer, 2004.

[36] D. Maoutsa, S. Reich, and M. Opper, “Interacting particle solutions of fokker–planck equations through gradient–log–density estimation,” *Entropy*, vol. 22, no. 8, p. 802, 2020.

[37] D. Onken, S. W. Fung, X. Li, and L. Ruthotto, “OT-flow: Fast and accurate continuous normalizing flows via optimal transport,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 9223–9232, 2021.

[38] R. J. LeVeque, *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems.* SIAM, 2007.

[39] G. Conforti, A. Durmus, and M. G. Silveri, “Score diffusion models without early stopping: finite Fisher information is all you need,” *arXiv preprint arXiv:2308.12240*, 2023.

[40] K. B. Petersen and M. S. Pedersen, “The matrix cookbook,” *Technical University of Denmark*, vol. 7, no. 15, p. 510, 2008.

[41] C.-H. Lai, Y. Takida, N. Murata, T. Uesaka, Y. Mitsufuji, and S. Ermon, “FP-diffusion: Improving score-based diffusion models by enforcing the underlying score fokker-planck equation,” in *International Conference on Machine Learning*, pp. 18365–18398, PMLR, 2023.

[42] J. C. Pinheiro and D. M. Bates, “Unconstrained parametrizations for variance-covariance matrices,” *Statistics and computing*, vol. 6, pp. 289–296, 1996.