Minimizing f-Divergences by Interpolating Velocity Fields

Song Liu [1] **Jiahao Yu** [1] **Jack Simons** [1] **Mingxuan Yi** [1] **Mark Beaumont** [1]

Abstract

Many machine learning problems can be seen as approximating a *target* distribution using a *particle* distribution by minimizing their statistical discrepancy. Wasserstein Gradient Flow can move particles along a path that minimizes the f-divergence between the target and particle distributions. To move particles, we need to calculate the corresponding velocity fields derived from a density ratio function between these two distributions. Previous works estimated such density ratio functions and then differentiated the estimated ratios. These approaches may suffer from overfitting, leading to a less accurate estimate of the velocity fields. Inspired by non-parametric curve fitting, we directly estimate these velocity fields using interpolation techniques. We prove that our estimators are consistent under mild conditions. We validate their effectiveness using novel applications on domain adaptation and missing data imputation. The code for reproducing our results can be found at `https://github.com/anewgithubname/gradest2`.

1. Introduction

Many machine learning problems can be formulated as minimizing statistical divergences between distributions, e.g., Variational Inference (Blei et al., 2017), Generative Modeling (Nowozin et al., 2016; Yi et al., 2023), Domain Adaptation (Courty et al., 2017a; Yu et al., 2021), and Data Imputation (Muzellec et al., 2020). Among many divergence minimization techniques, Particle-based gradient descent reduces the divergence between a set of particles (which define a distribution) and the target distribution by iteratively moving particles according to an update rule.

One such algorithm is Stein Variational Gradient Descent (SVGD) (Liu and Wang, 2016; Liu, 2017). It minimizes the Kullback-Leibler (KL) divergence using the steepest descent algorithm and has achieved promising results in Bayesian inference. However, to compute the SVGD updates, we need unnormalized target density functions. If we only have samples from the target distribution—as is often the case in applications like domain adaptation and generative model training— we cannot directly apply SVGD to minimize the f-divergences.

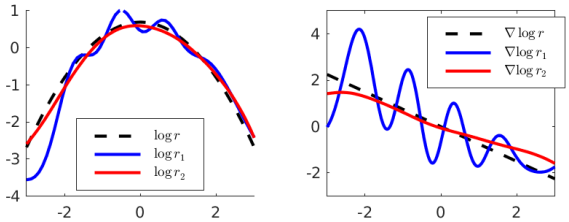


Figure 1. Estimating a log density ratio $\log r$ using a flexible model (RBF kernel) leads to a overfitted estimate ($\log r_1$). The overfitting consequently causes huge fluctuations in the derivative $(\log r_1)'$. Our proposed method provides a much more stable estimate $\log r_2$ and a more accurate estimate of $(\log r_2)'$.

Wasserstein Gradient Flow (WGF) describes the evolution of a marginal measure q_t along the steepest descent direction of a functional objective in Wasserstein geometry where $x_t \sim q_t$ follows a probability flow ODE. It can be employed to minimize an f-divergence between a particle distribution and the target distribution. Particularly, such WGFs characterize the following ODE (Yi et al., 2023; Gao et al., 2019)

$$\mathrm{d}\boldsymbol{x}_t = \nabla(h \circ r_t)(\boldsymbol{x}_t)\mathrm{d}t, \quad t \in [0, \infty).$$

Here $r_t := \frac{p}{q_t}$ is the ratio between the target density p and particle density q_t at time t, and h is a known function which depends on the f-divergence. The gradient operator ∇ computes the gradient of the composite function $h \circ r_t$. If we know r_t, simulating the above ODE would be straightforward using a discrete-time Euler method. However, the main challenge is that we do not know the ratio r_t in practice. In the context we consider, we know neither the particle nor target density, which constitutes r_t.

To overcome this issue, recent works (Gao et al., 2019; Ansari et al., 2021; Simons et al., 2021) first obtain an estimate $\hat{r}_t$ using a density ratio estimator (Sugiyama et al.,

[1]University of Bristol, Bristol, UK. Correspondence to: Song Liu <song.liu@bristol.ac.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

2012), then differentiate $h \circ \hat{r}_t$ to obtain $\nabla(h \circ \hat{r}_t)(\boldsymbol{x}_t)$. However, like other estimation tasks, density ratio estimation can be prone to overfitting. The risk of overfitting is further exacerbated when employing flexible models such as kernel models or neural networks. Overfitting can be disastrous for gradient estimation: A wiggly fit of r_t will cause huge fluctuations in gradient (see the blue fit in Figure 1). Moreover, density ratio estimation lacks the inductive bias for the density ratio gradient estimation. In other words, while it might be good at estimating the ratio itself, it doesn't have any built-in assumptions to capture the gradient of the density ratio function accurately.

In this paper, we **directly approximate the velocity fields induced by WGF**, i.e., $\nabla(h \circ r_t)(\boldsymbol{x}_t)$. We show that the backward KL velocity field, where $h = \log$, can be effectively estimated using Nadaraya-Watson (NW) interpolation if we know $\nabla \log p$, and this estimator is closely related to SVGD. We prove the estimation error of $\nabla \log r_t(\boldsymbol{x}_t)$ vanishes as the kernel bandwidth approaches to zero. This finding motivates us to propose a more general linear interpolation method to approximate $\nabla(h \circ r_t)(\boldsymbol{x}_t)$ for any general h functions using only samples from p and q_t. Our estimators are based on the idea that, within the neighbourhood of a given point, the best linear approximation of $h \circ r_t$ has slope $\nabla(h \circ r_t)$. Under mild conditions, we show that our estimators are also consistent for estimating $\nabla(h \circ r_t)(\boldsymbol{x}_t)$ and achieve the optimal non-parametric regression rate. Finally, equipped with our proposed gradient estimator, we test WGF on two novel applications: *domain adaptation* and *missing data imputation* and achieve promising performance.

2. Background

Notation: $\mathbb{R}^d$ is the d-dimensional real domain. Vectors are lowercase bold letters, e.g., $\boldsymbol{x} := [x_1, \ldots, x_d]^\top$. $\boldsymbol{x}_{-j}$ is a subvector of $\boldsymbol{x}$ obtained by excluding the j-th dimension. Matrices are uppercase bold letters, e.g., $\boldsymbol{X}, \boldsymbol{Y}$. $f \circ g$ is the composite function $f(g(\cdot))$. f' is the derivative of a univariate function. $\partial_i f$ means the partial derivative with respect to the i-th input of f and $\nabla f := [\partial_1 f, \partial_2 f, \cdots, \partial_d f]^\top$. $\nabla^\top f$ represents its transpose. $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{y})$ represents the gradient of f with respect to the input $\boldsymbol{x}$. $\nabla \boldsymbol{f} \in \mathbb{R}^{d \times m}$ represents the Jacobian of a vector-valued function $\boldsymbol{f} : \mathbb{R}^m \to \mathbb{R}^d$. $\partial_i^2 f$ means the second-order partial derivative with respect to the i-th input of f. $\nabla^2 f$ is the Hessian of f. $\|\cdot\|$ is the ℓ_2 norm of a vector or the spectral norm of a matrix. $\lambda_{\min}[\boldsymbol{X}]$ represents the smallest eigenvalue of a matrix $\boldsymbol{X}$. $a \vee b$ is the greater value between two scalars a and b. $\mathcal{P}(\mathbb{R}^d)$ is the space of probability measures defined on $\mathbb{R}^d$ equipped with Wasserstein-2 metric or Wasserstein space for short. $\hat{\mathbb{E}}[\cdot]$ is the sample approximation of an expectation $\mathbb{E}[\cdot]$.

We begin by introducing WGF of f-divergence, an effective technique for minimizing f-divergence between the particle and target distribution.

Name	Notation	$f(r_t)$	$h(r_t)$
Forw. KL	$\mathrm{KL}[p, q_t]$	$r_t \log(r_t)$	r_t
Back. KL	$\mathrm{KL}[q_t, p]$	$-\log(r_t)$	$\log r_t - 1$
Pearson's χ^2	$\chi^2_{\mathrm{p}}[p, q_t]$	$\frac{1}{2}(r_t - 1)^2$	$\frac{1}{2}r_t^2 - \frac{1}{2}$
Neyman's χ^2	$\chi^2_{\mathrm{n}}[p, q_t]$	$\frac{1}{2r_t} - \frac{1}{2}$	$-\frac{1}{r_t} + \frac{1}{2}$

Table 1. Some f-divergences, their definitions and h functions computed according to Theorem 2.1.

2.1. Wasserstein Gradient Flows of f-divergence

In general terms, a Wasserstein Gradient Flow is a curve in probability space (Ambrosio et al., 2005). By moving a probability measure along this curve, a functional objective (such as a statistical divergence) is reduced. In this work, we focus solely on using f-divergences as the functional objective. Let $q_t : \mathbb{R}_+ \to \mathcal{P}(\mathbb{R}^d)$ be a curve in Wasserstein space. Consider an f-divergence defined by $D_f[p, q_t] := \int q_t(\boldsymbol{x}) f(r_t(\boldsymbol{x})) \mathrm{d}\boldsymbol{x}$, where $r_t := \frac{p}{q_t}$. f is a twice differentiable convex function with $f(1) = 0$.

Theorem 2.1 (Corollary 3.3 in (Yi et al., 2023))**.** *The Wasserstein gradient flow of $D_f[p, q_t]$ characterizes the particle evolution via the ODE:*

$$\mathrm{d}\boldsymbol{x}_t = \nabla(h \circ r_t)(\boldsymbol{x}_t)\mathrm{d}t, \; h(r_t) = r_t f'(r_t) - f(r_t).$$

Simply speaking, particles evolve in Euclidean space according to the above ODE moves the corresponding q_t along a curve where $D_f[p, q_t]$ always decreases with time.

Theorem 2.1 establishes a relationship between f and a function $h : \mathbb{R}_+ \mapsto \mathbb{R}$. The gradient field of $h \circ r_t$ over time as $t \to \infty$ is referred to as the *WGF velocity field*. Similar theorems on reversed f-divergences have been discussed in (Gao et al., 2019; Ansari et al., 2021). Some frequently used f-divergences and their corresponding h functions are listed in Table 1. Specifically, for the backward KL divergence we have $\nabla(h \circ r_t)(\boldsymbol{x}_t)|_{h=\log(\cdot)} = \nabla \log r_t(\boldsymbol{x}_t)$.

In reality, we move particles by simulating the above ODE using the forward Euler method: We draw particles from an initial distribution q_0 and iteratively update them for time $t = 0, 1 \ldots T$ according to the following rule:

$$\boldsymbol{x}_{t+1} := \boldsymbol{x}_t + \eta \nabla(h \circ r_t)(\boldsymbol{x}_t) \tag{1}$$

where η is a small step size. There is a slight abuse of notation, and we reuse t for discrete-time indices [1].

Although (1) seems straightforward, we normally do not have access to r_t, so the update in (1) cannot be readily performed. In previous works, such as (Gao et al., 2019;

[1] From now on, we will only discuss discrete-time algorithms.

Simons et al., 2021), r_t is estimated using density ratio estimators and the WGF is simulated using the estimated ratio. Although these estimators achieved promising results, the density ratio estimators are not designed for usage in WGF algorithms. For example, a small density ratio estimation error could lead to huge deviations in gradient estimation, as we demonstrated in Figure 1. Others (Wang et al., 2022) propose to estimate the gradient flow using Kernel Density Estimation (KDE) on densities p and q_t separately (See Section J), then compute the ratio. However, KDE tends to perform poorly in high dimensional settings (see e.g., (Scott, 1991)).

3. Direct Velocity Field Estimation by Interpolation

In this work, we consider directly estimating the velocity field, i.e., directly modelling and estimating $\nabla(h \circ r_t)$. We are encouraged by the recent successes in Score Matching (Hyvärinen, 2005; 2007; Vincent, 2011; Song et al., 2020), which is a direct estimator of a log density gradient. It works by minimizing the squared differences between the true log density gradient and the model gradient. However, such a technique cannot be easily adapted to estimate $\nabla(h \circ r_t)$, even for $h(r) = \log r$ (See Appendix K).

We start by looking at a simpler setting where $\nabla \log p$ is known. In fact, this setting itself has many interesting applications such as Bayesian inference. The solution we derive using interpolation will serve as a motivation for other interpolation based approaches in later sections.

3.1. Nadaraya-Watson (NW) Interpolation of Backward KL Velocity Field

Define a local weighting function with a parameter $\sigma > 0$, $k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star) := \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}^\star\|^2}{2\sigma^2}\right)$. Nadaraya-Watson (NW) estimator (Nadaraya, 1964; Watson, 1964) interpolates a function g at **a fixed point** $\boldsymbol{x}^\star$. Suppose that we observe $g(\boldsymbol{x})$ at a set of sample points $\{\boldsymbol{x}_i\}_{i=1}^n \sim q$, NW interpolates $g(\boldsymbol{x}^\star)$ by computing

$$\hat{g}(\boldsymbol{x}^\star) := \widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star) g(\boldsymbol{x})] / \widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)]. \quad (2)$$

Thus, the NW interpolation of the backward KL field [2] is

$$\hat{\boldsymbol{u}}_t(\boldsymbol{x}^\star) := \widehat{\mathbb{E}}_{q_t}[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star) \nabla \log r_t(\boldsymbol{x})] / \widehat{\mathbb{E}}_{q_t}[k(\boldsymbol{x}, \boldsymbol{x}^\star)]. \quad (3)$$

Since we cannot evaluate $\nabla \log r_t(\boldsymbol{x})$, (3) is intractable. However, assuming that $\lim_{\|\boldsymbol{x}\| \to \infty} q_t(\boldsymbol{x}) k(\boldsymbol{x}, \boldsymbol{x}^\star) = 0$, using integration by parts , the expectation in the numerator of (3) can be rewritten as:

$$\begin{aligned}\mathbb{E}_{q_t}\left[k_\sigma^\star \nabla \log r_t(\boldsymbol{x})\right] =& \mathbb{E}_{q_t}\left[k_\sigma^\star \nabla \log p(\boldsymbol{x})\right] - \mathbb{E}_{q_t}\left[\nabla \log q_t(\boldsymbol{x})\right] \\ =& \mathbb{E}_{q_t}[k_\sigma^\star \nabla \log p(\boldsymbol{x}) + \nabla k_\sigma^\star], \quad (4)\end{aligned}$$

where we shortened the kernel $k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)$ as $k_\sigma^\star$. Since we can evaluate $\nabla \log p$, (4) can be approximated using samples from the particle distribution q_t. Thus, the NW estimator of the backward KL field can be approximated by

$$\hat{\boldsymbol{u}}_t(\boldsymbol{x}^\star) \approx \widehat{\mathbb{E}}_{q_t}[k_\sigma^\star \nabla \log p(\boldsymbol{x}) + \nabla k_\sigma^\star] / \widehat{\mathbb{E}}_{q_t}[k_\sigma^\star]. \quad (5)$$

Interestingly, the numerator of (5) is exactly the particle update of the SVGD algorithm (Liu and Wang, 2016) for an RKHS induced by a Gaussian kernel (See Appendix N for details on SVGD), and the equality (4) has been noticed by Chewi et al. (2020). Note that for different $\boldsymbol{x}^\star$, the denominator in (5) is different and thus cannot be combined into the overall learning rate of SVGD.

3.2. Effectiveness of NW Estimator

For simplicity, we drop t from $\boldsymbol{u}_t$, r_t, $\boldsymbol{x}_t$ and q_t when our analysis holds the same for all t.

Although there have been theoretical justifications for the convergence analysis of WGF given the ground truth velocity fields such as Langevin dynamics (Wibisono, 2018). Few theories have been dedicated to the estimation of velocity fields themselves. One of the contributions of this paper is that we study the statistical theory of the velocity field estimation through the lenses of non-parametric regression/curve approximation.

Now, we prove the convergence rate of the NW estimator under the assumption that the second-order derivative of $\log r$ is well-behaved. Although $\hat{\boldsymbol{u}}(\boldsymbol{x}^\star)$ cannot be directly computed, assuming $\nabla \log p$, k and ∇k are well-behaved, using concentration inequalities (such as Hoeffding's inequality (Hoeffding, 1963)), the difference between $\hat{\boldsymbol{u}}(\boldsymbol{x}^\star)$ and its approximation (5) can be easily bounded. Thus, we focus on the classical NW estimator in (3).

Proposition 3.1. *Suppose* $\sup_{\boldsymbol{x} \in \mathbb{R}^d} \|\nabla^2 \log r(\boldsymbol{x})\| \le \kappa < \infty$. *Define* $k(\boldsymbol{y}) := \exp\left[-\|\boldsymbol{y}\|^2/2\right]$. *Assume that there exist constants* C_k, K *that are independent of* σ, *such that*

$$\frac{\int q(\sigma \boldsymbol{y} + \boldsymbol{x}^\star) k(\boldsymbol{y}) \|\boldsymbol{y}\| \mathrm{d}\boldsymbol{y}}{\int q(\sigma \boldsymbol{y} + \boldsymbol{x}^\star) k(\boldsymbol{y}) \mathrm{d}\boldsymbol{y}} \le C_k, \mathbb{E}_q\left[\frac{1}{\sigma^d} k_\sigma^\star\right] \ge K > 0 \quad (6)$$

$$\mathrm{Var}_q\left[\frac{1}{\sigma^d} k_\sigma^\star \|\boldsymbol{x} - \boldsymbol{x}^\star\|\right] = O\left(\frac{1}{\sigma^d}\right),$$
$$\mathrm{Var}_q\left[\frac{1}{\sigma^d} k_\sigma^\star\right] = O\left(\frac{1}{\sigma^d}\right), \text{ as } \sigma \to 0. \quad (7)$$

Then, with high probability, there exists a constant K':

$$\|\hat{\boldsymbol{u}}(\boldsymbol{x}^\star) - \nabla \log r(\boldsymbol{x}^\star)\| \le \sqrt{d}\kappa \left[\frac{K'}{\sqrt{n\sigma^d}} + \sigma C_k\right]$$

[2] "backward KL field" is short for backward KL velocity field.

for all $\sigma > 0$.

We can also deduce that, with an optimal choice of $\sigma \sim n^{-1/(d+2)}$, the estimation error is $O_p(n^{-1/(d+2)})$.

See Appendix A for the proof. $q(\sigma \boldsymbol{y} + \boldsymbol{x}^\star)$ in the first inequality (6) can be further expanded using Taylor expansion. Provided that the kernel k is well-behaved, this condition becomes a regularity condition on $q(\boldsymbol{x}^\star)$ and its higher order moments. The second inequality in (6) means that there should be enough mass around $\boldsymbol{x}^\star$ under the distribution q, which is a key assumption in classical nonparametric curve estimation (See, e.g., Chapter 20 in (Wasserman, 2010)). (7) is required to ensure the empirical quantities in the NW estimator converge to their population counterparts in probability.

It can be seen that the estimation error is bounded by the sample approximation error $\frac{K'}{\sqrt{n\sigma^d}}$ and a bias depending on by σ. Interestingly, the bias term decreases at the rate of σ, slower than the classical rate σ^2 for the non-parametric regression of a second-order differentiable function (Theorem 20.21 in (Wasserman, 2010)). This is expected as NW estimates the *gradient* of $h \circ r$, not $h \circ r$. To achieve a faster σ^2 rate, one needs to assume conditions on $\nabla^3 \log r$. This also highlights a slight downside of using NW to estimate the gradient. However, in the following section, we show that another interpolator achieves the superior rate when using the same type of assumption on $\nabla^2(h \circ r)$.

4. Velocity Field Interpolation from Samples

Although we have seen that $\hat{\boldsymbol{u}}$ is an effective estimator of the backward KL velocity field, it can only be approximated when we can evaluate $\nabla \log p$. In some applications, such as domain adaptation or generative modelling, we only have samples from the target distribution, and $\nabla \log p$ is unavailable. Moreover, since the f-divergence family consists of a wide variety of divergences, we hope to provide *a general computational framework* to estimate different velocity fields that minimize different f-divergences.

Nonetheless, the success of NW motivates us to look for other interpolators to approximate $\nabla(h \circ r)(\boldsymbol{x}^\star)$.

Another common interpolation technique is *local linear regression* (See e.g., (Gasser and Müller, 1979; Fan, 1993) or Chapter 6, (Hastie et al., 2001)). It approximates an unknown function g at $\boldsymbol{x}^\star$ by using a linear function: $\hat{g}(\boldsymbol{x}) := \langle \boldsymbol{\beta}(\boldsymbol{x}^\star), \boldsymbol{x} \rangle + \beta_0(\boldsymbol{x}^\star)$. $\boldsymbol{\beta}(\boldsymbol{x}^\star)$ and $\beta_0(\boldsymbol{x}^\star)$ are the minimizer of the following weighted least squares objective:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^d, \beta \in \mathbb{R}} \widehat{\mathbb{E}}_q \left[k_\sigma^\star \left(g(\boldsymbol{x}) - \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle - \beta_0 \right)^2 \right]. \tag{8}$$

A key insight is, since *the gradient of a function is the slope of its best local linear approximation*, it is reasonable to just use the slope of the fitted linear model, a.k.a., $\boldsymbol{\beta}(\boldsymbol{x}^\star)$, to approximate the gradient $\nabla g(\boldsymbol{x}^\star)$. See Figure 6 in Appendix for an illustration.

We apply the same rationale to estimate $\nabla(h \circ r)(\boldsymbol{x}^\star)$. However, unlike local linear interpolation, we cannot evaluate $h \circ r$ at any input. Thus, we cannot directly use the least squares objective (8) to obtain a local linear interpolation. Similar to what we have done in Section 3.1, we look for a tractable population estimator for estimating $h \circ r$, which can be approximated using samples from p and q. Then, we "convert it" into a local linear objective.

In the following section, we derive an objective for estimating $h \circ r$ by maximizing a variational lower bound of a *mirror divergence*.

4.1. Mirror Divergence

Definition 4.1. Let $D_\phi[p, q]$ and $D_\psi[p, q]$ denote two f-divergences with f being ϕ and ψ respectively. D_ψ is the mirror of D_ϕ if and only if $\psi'(r) \triangleq r\phi'(r) - \phi(r)$, where $\triangleq$ means equal up to a constant.

For example, let $D_\phi[p, q]$ and $D_\psi[p, q]$ be $\mathrm{KL}[p, q]$ and $\chi^2_{\mathrm{p}}[p, q]$ respectively. From Table 1, we can see that $\phi(r) = r \log r$ and $\psi(r) = \frac{1}{2}(r-1)^2$. Thus $r\phi'(r) - \phi(r) = r(1 + \log r) - r \log r = r \triangleq \psi'(r) = r - 1$. Therefore, $\chi^2_{\mathrm{p}}[p, q]$ is the mirror of $\mathrm{KL}[p, q]$. Similarly, we can verify that $\mathrm{KL}[p, q]$ is the mirror of $\mathrm{KL}[q, p]$ and $\mathrm{KL}[q, p]$ is the mirror of $\chi^2_{\mathrm{n}}[p, q]$. In general, $D_\psi[p, q]$ is the mirror of $D_\phi[p, q]$ does not imply the other direction.

4.2. Gradient Estimator using Linear Interpolation

The key observation that helps derive a tractable objective is that $h \circ r$ is the argmax of "the mirror variational lowerbound". Suppose h is associated with an f-divergence D_ϕ as per Theorem 2.1 and D_ψ is the mirror of D_ϕ. Then $h \circ r$ is the argmax of the following objective:

$$D_\psi[p, q] = \max_d \mathbb{E}_p[d(\boldsymbol{x})] - \mathbb{E}_q[\psi_{\mathrm{con}}(d(\boldsymbol{x}))], \tag{9}$$

where ψ_{con} is the *convex conjugate* of ψ. The formal statement and its proof can be found in Appendix C. The equality in (9) is known in previous literature (Nguyen et al., 2010; Nowozin et al., 2016) and the objective in (9) is commonly referred to as the variational lowerbound of $D_\psi[p, q]$.

Notice that the expectations in (9) can be approximated by $\widehat{\mathbb{E}}_p[\cdot]$ and $\widehat{\mathbb{E}}_q[\cdot]$ using samples from p and q respectively.

This is a *surprising result*. Since h is related to D_ϕ's field (as per Theorem 2.1), one may associate maximizing D_ϕ's variational lowerbound with its velocity field estimation. However, the above observation shows that, to approximate D_ϕ's field, one should maximize the variational lowerbound

of its *mirror divergence* D_ψ! To our best knowledge, this "mirror structure" in the context of WGF has never been studied before.

We then localize (9) to obtain a local linear estimator of $h \circ r$ at a fixed point $\boldsymbol{x}^\star$. First, we parameterize the function d using a linear model $d_{\boldsymbol{w},b}(\boldsymbol{x}) := \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b$. Second, we weight the objective using $k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)$, which leads to the following local linear objective:

$$(\boldsymbol{w}(\boldsymbol{x}^\star), b(\boldsymbol{x}^\star)) = \operatorname*{argmax}_{\boldsymbol{w}\in\mathbb{R}^d, b\in\mathbb{R}} \ell(\boldsymbol{w}, b; \boldsymbol{x}^\star),$$

$$\text{where } \ell(\boldsymbol{w}, b; \boldsymbol{x}^\star) := \widehat{\mathbb{E}}_p[k_\sigma^\star d_{\boldsymbol{w},b}(\boldsymbol{x})] - \widehat{\mathbb{E}}_q[k_\sigma^\star \psi_{\text{con}}(d_{\boldsymbol{w},b}(\boldsymbol{x}))] \tag{10}$$

The above transformation is similar to how the local linear regression "localizes" the ordinary least squares objective.

Solving (10), we get a linear approximation of $h \circ r$ at $\boldsymbol{x}^\star$:

$$h(r(\boldsymbol{x}^\star)) \approx \langle \boldsymbol{w}(\boldsymbol{x}^\star), \boldsymbol{x}^\star \rangle + b(\boldsymbol{x}^\star).$$

Following the intuition that $\nabla(h \circ r)(\boldsymbol{x}^\star)$ is the slope of the best local linear fit of $h(r(\boldsymbol{x}^\star))$, we use $\boldsymbol{w}(\boldsymbol{x}^\star)$ to approximate $\nabla(h \circ r)(\boldsymbol{x}^\star)$. We will theoretically justify this approximation in Section 4.3. Now let us study two examples:

Example 4.2. Suppose we would like to estimate $\mathrm{KL}[p, q]$'s field at $\boldsymbol{x}^\star$, which is $\nabla r(\boldsymbol{x}^\star)$. Using Definition 4.1, we can verify that the mirror of $\mathrm{KL}[p, q]$ is $D_\psi = \chi^2_{\mathrm{p}}[p, q]$, in which case $\psi = \frac{1}{2}(r-1)^2$. The convex conjugate of ψ is $\psi_{\text{con}}(d) = d^2/2 + d$. Substituting ψ_{con} in (10), the gradient estimator $\boldsymbol{w}_\rightarrow(\boldsymbol{x}^\star) \approx \nabla r(\boldsymbol{x}^\star)$ is obtained by the following objective:

$$(\boldsymbol{w}_\rightarrow(\boldsymbol{x}^\star), b_\rightarrow(\boldsymbol{x}^\star)) := \operatorname*{argmax}_{\boldsymbol{w}\in\mathbb{R}^d, b\in\mathbb{R}} \ell_\rightarrow(\boldsymbol{w}, b; \boldsymbol{x}^\star)$$

$$\text{where } \ell_\rightarrow(\boldsymbol{w}, b; \boldsymbol{x}^\star) := \widehat{\mathbb{E}}_p[k_\sigma^\star \cdot d_{\boldsymbol{w},b}(\boldsymbol{x})] - \widehat{\mathbb{E}}_q\left[k_\sigma^\star \cdot \left(\frac{d_{\boldsymbol{w},b}(\boldsymbol{x})^2}{2} + d_{\boldsymbol{w},b}(\boldsymbol{x})\right)\right]. \tag{11}$$

Example 4.3. Suppose we would like to estimate $\mathrm{KL}[q, p]$'s field at $\boldsymbol{x}^\star$, which is $\nabla \log r(\boldsymbol{x}^\star)$. Using Definition 4.1, we can verify that the mirror of $\mathrm{KL}[q, p]$ is $D_\psi = \mathrm{KL}[p, q]$, in which case, $\psi(r) = r \log r$. The convex conjugate of ψ is $\psi_{\text{con}}(d) = \exp(d-1)$. The gradient estimator $\boldsymbol{w}_\leftarrow(\boldsymbol{x}^\star) \approx \nabla \log r(\boldsymbol{x}^\star)$ is obtained by the following objective:

$$(\boldsymbol{w}_\leftarrow(\boldsymbol{x}^\star), b_\leftarrow(\boldsymbol{x}^\star)) := \operatorname*{argmax}_{\boldsymbol{w}\in\mathbb{R}^d, b\in\mathbb{R}} \ell_\leftarrow(\boldsymbol{w}, b; \boldsymbol{x}^\star)$$

$$\text{where } \ell_\leftarrow(\boldsymbol{w}, b; \boldsymbol{x}^\star) := \widehat{\mathbb{E}}_p[k_\sigma^\star \cdot d_{\boldsymbol{w},b}(\boldsymbol{x})] - \widehat{\mathbb{E}}_q[k_\sigma^\star \cdot \exp(d_{\boldsymbol{w},b}(\boldsymbol{x}) - 1)]. \tag{12}$$

In the following section, we show that the estimation error $\|\boldsymbol{w}(\boldsymbol{x}^\star) - \nabla(h \circ r)(\boldsymbol{x}^\star)\|$ vanishes as $\sigma \to 0$ and $n \to \infty$.

4.3. Effectiveness of Local Interpolation

In this section, we state our main theoretical result. Let $(\boldsymbol{w}(\boldsymbol{x}^\star), b(\boldsymbol{x}^\star))$ be a stationary point of $\ell(\boldsymbol{w}, b; \boldsymbol{x}^\star)$. We denote the domain of $\boldsymbol{x}$ as $\mathcal{X}$ (not necessarily $\mathbb{R}^d$). Without loss of generality, we also assume all sample averages $\widehat{\mathbb{E}}[\cdot]$ are averaged over n samples. We prove that, $\boldsymbol{w}(\boldsymbol{x}^\star)$ is a consistent estimate of $\nabla(h \circ r)(\boldsymbol{x}^\star)$ assuming the change rate of the flow is bounded.

Assumption 4.4. The change rate of the velocity fields is well-behaved, i.e.,

$$\sup_{\boldsymbol{x}\in\mathcal{X}} \|\nabla^2(h \circ r)(\boldsymbol{x})\| \le \kappa.$$

This is an analogue of the assumption on $\nabla^2 \log r$ in Proposition 3.1.

Assumption 4.5. There exists a constant $C_k > 0$ independent of σ,

$$\frac{1}{2}\int q(\sigma\boldsymbol{y} + \boldsymbol{x}^\star) k(\boldsymbol{y}) \cdot \|\boldsymbol{y}\|^2 \cdot \|[\sigma\boldsymbol{y} + \boldsymbol{x}^\star, 1]\| \,\mathrm{d}\boldsymbol{y} \le C_k.$$

This assumption is similar to the first inequality in (6). Expanding $q(\sigma\boldsymbol{y} + \boldsymbol{x}^\star)$ using Taylor expansion and providing that our kernel is well behaved, this assumption essentially implies the boundedness of the $q(\boldsymbol{x}^\star)$ and $\|\nabla^2 q(\boldsymbol{x}^\star)\|$.

Define two shorthands: $\boldsymbol{w}^* := \nabla(h \circ r)(\boldsymbol{x}^\star)$ and $b^* := h(r(\boldsymbol{x}^\star)) - \langle \nabla(h \circ r)(\boldsymbol{x}^\star), \boldsymbol{x}^\star \rangle$.

Assumption 4.6. Let $\tilde{\boldsymbol{x}} := \left[\boldsymbol{x}^\top, 1\right]^\top$. As $\sigma \to 0$,

$$\operatorname{tr}\left[\operatorname{Cov}_p\left[\frac{1}{\sigma^d} k_\sigma^\star \cdot \tilde{\boldsymbol{x}}\right]\right] = O(\frac{1}{\sigma^d}),$$

$$\operatorname{tr}\left[\operatorname{Cov}_q\left[\frac{1}{\sigma^d} k_\sigma^\star \cdot \psi'_{\text{con}}(\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle + b^*)\tilde{\boldsymbol{x}}\right]\right] = O(\frac{1}{\sigma^d}).$$

These two are analogues of (7). They are required so that our sample approximation of the objective is valid and concentration inequalities can be applied.

Assumption 4.7. For all $a \in [0, 1]$ and $\boldsymbol{x} \in \mathcal{X}$, $\psi''_{\text{con}}\left[a h(r(\boldsymbol{x})) + (1-a)(\langle \boldsymbol{w}^*, \boldsymbol{x} \rangle + b^*)\right] \le C_{\psi''_{\text{con}}}$.

This is a unique assumption to our estimator, where we assume that the convex conjugate ψ_{con} has a bounded second-order derivative.

Theorem 4.8. *Suppose Assumption 4.4, 4.5, 4.6 and 4.7 holds. If there exist strictly positive constants $W, B, \Lambda_{\min}$ that are independent of σ and n such that,*

$$\|\boldsymbol{w}^*\| \le W, \quad |b^*| \le B \tag{13}$$

and for all $\boldsymbol{w} \in \{\boldsymbol{w} | \|\boldsymbol{w}\| < 2W\}$ and $b \in \{b | |b| < 2B\}$,

$$\lambda_{\min}\left\{\widehat{\mathbb{E}}_q\left[k_\sigma^\star \nabla^2_{[\boldsymbol{w},b]} \psi_{\text{con}}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b)\right]\right\} \ge \sigma^d \Lambda_{\min}, \tag{14}$$

holds with high probability. Then there exists $\sigma_0, N, K > 0$ such that for all $0 < \sigma < \sigma_0, n > N$,

$$\|\boldsymbol{w}(\boldsymbol{x}^\star) - \boldsymbol{w}^*\| \leq \frac{\frac{K}{\sqrt{n\sigma^d}} + \kappa C_k C_{\psi''_{\text{con}}} \sigma^2}{\Lambda_{\min}},$$

with high probability.

The proof can be found in Appendix D.

Similar to Proposition 3.1, the estimation error is upper-bounded by the sample approximation error that reduces with the "effective sample size" $n\sigma^d$, and a bias term that reduces with σ. Interestingly, although we make the smoothness assumption on $\nabla^2(h \circ r)$, similar to Proposition 3.1, the bias vanishes at a *quadratic rate* σ^2, unlike the linear rate obtained in Proposition 3.1.

We can deduce that, with an optimal choice of $\sigma \sim n^{-1/(d+4)}$, the estimation error is $O_p(n^{-2/(d+4)})$, a rate faster than the rate implied by Proposition 3.1.

Using Theorem 4.8, we can prove the consistency of various velocity field estimators for different f-divergences.

Corollary 4.9. *Suppose $\sup_{\boldsymbol{x}\in\mathcal{X}} \|\nabla^2 r(\boldsymbol{x})\| \leq \kappa_{\rightarrow}$, Assumption 4.5, 4.6 holds and*

$$\lambda_{\min}\left\{\widehat{\mathbb{E}}_q\left[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\top\right]\right\} \geq \sigma^d \cdot \Lambda_{\rightarrow} > 0, \qquad (15)$$

holds with high probability. Then $\exists \sigma_0, N, K > 0$,

$$\|\boldsymbol{w}_{\rightarrow}(\boldsymbol{x}^\star) - \nabla r(\boldsymbol{x}^\star)\| \leq \frac{\frac{K}{\sqrt{n\sigma^d}} + \kappa_{\rightarrow} \cdot C_k \cdot \sigma^2}{\Lambda_{\rightarrow}}$$

for all $0 < \sigma < \sigma_0, n > N$ holds with high probability.

See Appendix E for the proof.

Corollary 4.10. *Suppose $\sup_{\boldsymbol{x}\in\mathcal{X}} \|\nabla^2 \log r(\boldsymbol{x})\| \leq \kappa_{\leftarrow}$, and Assumption 4.5, 4.6 holds. Assume*

$$\sup_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{x}\| \leq C_{\mathcal{X}}, \sup_{\boldsymbol{x}\in\mathcal{X}} \log r(\boldsymbol{x}) - 1 \leq C_{\log r},$$

and there exists $B, W < \infty$, so that

$$\|\nabla \log r(\boldsymbol{x}^\star)\| < W, |\log r(\boldsymbol{x}^\star) - \langle \nabla \log r(\boldsymbol{x}^\star), \boldsymbol{x}^\star\rangle| < B.$$

Additionally,

$$\lambda_{\min}\left[\widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\top]\right] \geq \sigma^d \cdot \Lambda_{\leftarrow} > 0, \qquad (16)$$

holds with high probability. Then $\exists \sigma_0, N, K > 0$ and for all $0 < \sigma < \sigma_0, n > N$,

$$\begin{aligned}&\|\boldsymbol{w}_{\leftarrow}(\boldsymbol{x}^\star) - \nabla \log r(\boldsymbol{x}^\star)\| \\ \leq &\frac{1}{(\Lambda_{\leftarrow} \cdot \exp(-2WC_{\mathcal{X}} - 2B - 1))} \cdot \frac{K}{\sqrt{n\sigma^d}} + \\ &\frac{\kappa_{\leftarrow} \cdot C_k \cdot [\exp(WC_{\mathcal{X}} + B - 1) \vee \exp(C_{\log r} - 1)] \cdot \sigma^2}{\Lambda_{\leftarrow} \cdot \exp(-2WC_{\mathcal{X}} - 2B - 1)},\end{aligned}$$

holds with high probability.

See Appendix F for the proof. Note that due to the assumption that $\|\boldsymbol{x}\|$ is bounded, Corollary 4.10 can only be applied to density ratio functions with bounded input domains. However, this does include important examples such as images, where pixel brightnesses are bounded within $[0, 1]$. Using the same proof techniques, it is possible to derive more Corollaries for other f-divergence velocity fields. We leave them as a future investigation.

4.4. Model Selection via Linear Interpolation

Although Theorem 4.8 says the estimation bias disappears as $\sigma \to 0$, when we only have a finite number of samples, the choice of the kernel bandwidth σ controls the bias-variance trade-off of the local estimation. Thus we propose a model selection criterion. The details of the procedure is provided in Appendix I.1.

The high level idea is: Suppose we have testing samples from p and q. A good choice of σ would result in a good approximation of $h \circ r$ on *testing points*, thus the best approximation of $h \circ r$ would maximize the variational lower bound (9). Therefore, we only need to evaluate (9) on testing samples to determine the optimality of σ.

Our local linear estimator offers a unique advantage of model selection because it is formulated as a non-parametric curve fitting problem. In contrast, SVGD lacks a systematic approach and has to resort to the "median trick".

5. Experiments

5.1. Reducing KL Divergence: SVGD vs. NW vs. Local Linear Estimator

In this experiment, we investigate the performance of SVGD, NW and Local Linear (LL) estimator through the task of minimizing $\mathrm{KL}[q_t, p]$. We let SVGD, NW and LL fit the target distribution $p = \mathcal{N}(0, 1)$. 500 iid initial particles are drawn from $q_0 = N(-1, 0.25^2)$. For all methods, we use naive gradient descent to update particles with a fixed step size 0.1. We also consider a variant of SVGD where AdaGrad (Duchi et al., 2011) is applied to adjust the step size dynamically. For SVGD and SVGD with AdaGrad, we use the MATLAB code provided by Liu and Wang (2016) with its default heuristics. We plot the trajectories of particles of all three methods in Figure 2.

Although all three algorithms move particles toward the target distribution, the naive SVGD does not spread the particle mass quickly enough to cover the target distribution when using the same step size. This situation is much improved by applying the adaptive learning rate method (AdaGrad). In comparison, NW and LL both converge fast. After 20 iterations, all particles have arrived at the target positions. Since all methods are motivated by minimizing

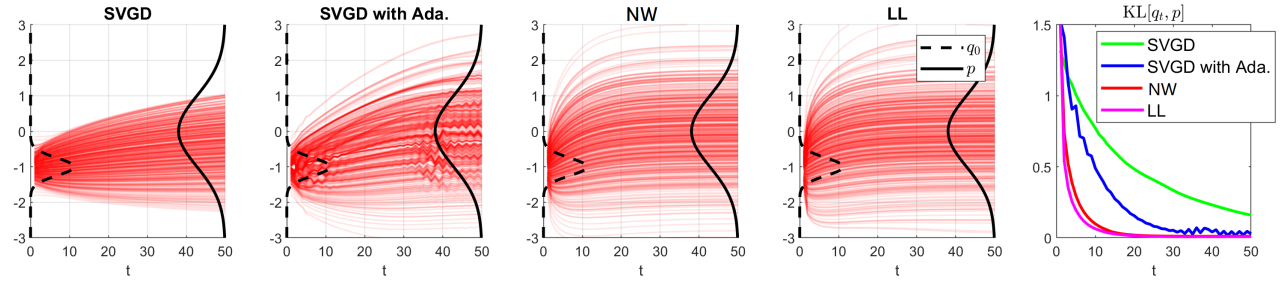


Figure 2. Particle Trajectories of SVGD, SVGD with AdaGrad, NW, LL. Approximated $\mathrm{KL}[q_t, p]$ with different methods.

$\mathrm{KL}[q_t, p]$, we plot the $\mathrm{KL}[q_t, p]$ approximated by Donsker and Varahan Lower Bound (Donsker and Varadhan, 1976) for all four methods. The plot of $\mathrm{KL}[q_t, p]$ agrees with our qualitative assessments: AdaGrad SVGD can reduce the KL significantly faster than the vanilla SVGD with naive gradient descent. After 20 iterations, the KL divergence for both NW and LL particles reaches zero, indicating that the particles have fully converged to the target distribution. LL achieves a performance comparable to NW. This is a remarkable result as NW, and SVGD has access to the true $\nabla \log p$, but LL only has samples from p.

In the next sections, we will showcase the performance of LL in forward/backward KL minimization problems.

5.2. Joint Domain Adaptation

In domain adaptation, we want to use source domain samples to help a prediction task in a target domain. This addresses situations where the training data for a method may differ from the real data when deployed. We assume that source samples $\mathcal{D}_q := \left\{(\boldsymbol{x}_q^{(i)}, y_q^{(i)})\right\}_{i=1}^{n_q}$ are drawn from a joint distribution $\mathbb{Q}_{XY}$ and target samples $\mathcal{D}_p := \left\{(\boldsymbol{x}_p^{(i)}, y_p^{(i)})\right\}_{i=1}^{n_p}$ are drawn from a different joint distribution $\mathbb{P}_{XY}$. However, y_p is missing from the target set. Thus, we want to predict missing labels in $\mathcal{D}_p$ with the help of $\mathcal{D}_q$. Courty et al. (2017b;a) propose to find an optimal map that aligns the distribution $\mathbb{Q}_{XY}$ and $\mathbb{P}_{XY}$, then train a classifier on the aligned source samples. Inspired by this method, we propose to align samples by minimizing $\mathrm{KL}[q_t, p]$, where p is the density of the target $\mathbb{P}_{XY}$ and q_t is a particle-label pair distribution whose samples are $\mathcal{D}_{q_t} := \left\{(\boldsymbol{x}_t^{(i)}, y_q^{(i)})\right\}_{i=1}^{n_q}$. To minimize $\mathrm{KL}[q_t, p]$, we evolve $\boldsymbol{x}_t$ according to the backward KL field and $\boldsymbol{x}_{t=0}$ is initialized to be the source input $\boldsymbol{x}_q$. In words, we transport source input samples so that the transported and target samples are aligned in terms of minimizing the backward KL divergence. After T iterations, we can train a classifier using transported source samples $\{(\boldsymbol{x}_T, y_q)\}$ to predict target labels.

One slight issue is that we do not have labels in the target domain but performing WGF requires joint samples (X, Y). To solve this, we adopt the same approach used in Courty et al. (2017a), replacing y_p with a proxy $\hat{g}(\boldsymbol{x}_p)$, where $\hat{g}$ is a prediction function trained to minimize an empirical transportation cost (See Section 2.2 in Courty et al. (2017a)). We demonstrate our approach in a toy example, in Figure 3.

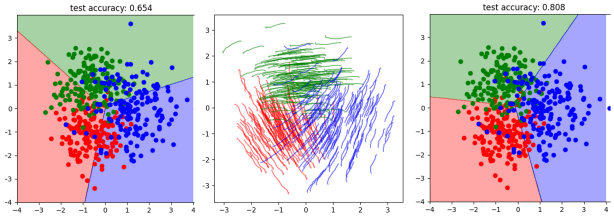


Figure 3. Left: the source classifier (represented by colored areas) misclassifies many testing points (colored dots). Middle: WGF moves particles to align the source and target samples. Lines are trajectories of sample movements in each class. Right: the retrained classifier on the transported source samples gives a much better prediction.

Table 2 compares the performance of adapted classifiers on a real-world 10-class classification dataset named "office-caltech-10", where images of the same objects are taken from four different domains (amazon, caltech, dslr and webcam). We reduce the dimensionality by projecting all samples to a 100-dimensional subspace using PCA. We compare the performance of the **base** (the source RBF kernel SVM classifier), the Joint Distribution Optimal Transport (Courty et al., 2017a) (**JDOT**), an RBF kernel SVM trained on the WGF transported source samples $\{(\boldsymbol{x}_T, y_q)\}$ (**WGF**) and an SVM trained on MMDFlow (Hagemann et al., 2024) transported source samples (**MMD**). The classification accuracy on the entire target sets are reported. It can be seen that in some cases, reusing the source classifiers in the target domain does lead to catastrophic results (e.g. amazon to dslr, caltech to dslr). However, we can avoid such performance decline by using any joint distribution-based domain adaptation. It can be seen that both **WGF** and **MMD** achieve

$\mathcal{D}_q \to \mathcal{D}_p$	base	JDOT	WGF	MMD
amz. → dslr	27.39%	65.61%	**78.34**%	**78.34**%
amz. → web.	61.69%	67.80%	84.07%	**89.15**%
amz. → cal.	81.66%	63.58%	**82.72**%	82.19%
dslr → amz.	70.35%	72.96%	**85.91**%	76.30%
dslr → web.	94.92%	76.61%	**95.25**%	86.10%
dslr → cal.	58.95%	72.31%	**79.25**%	69.01%
web. → amz.	75.78%	75.37%	**91.34**%	89.46%
web. → dslr	94.27%	73.25%	98.73%	**100.00**%
web. → cal.	67.05%	63.49%	**78.09**%	75.16%
cal. → amz.	83.40%	84.55%	**91.13**%	89.04%
cal. → dslr	26.11%	69.43%	**84.71**%	**84.71**%
cal. → web.	63.39%	74.58%	**80.34**%	79.32%

Table 2. Domain adaptation of different domains in Office-Caltech-10 dataset.

superior performance compared to **JDOT** while **WGF** has the best performance in most adaptation cases.

5.3. Missing Data Imputation

In missing data imputation, we are given a joint dataset $\tilde{\mathcal{D}} := \{(\tilde{\boldsymbol{x}}^{(i)}, \boldsymbol{m}^{(i)})\}$, where $\boldsymbol{m} \in \{0,1\}^d$ is a mask vector and $m_j^{(i)} = 0$ indicates the j-th dimension of $\tilde{\boldsymbol{x}}^{(i)}$ is missing. The task is to "guess" the missing values in $\tilde{\boldsymbol{x}}$ vector. In recent years, GAN-based missing value imputation, e.g., **GAIN** (Yoon et al., 2018), has gained significant attention (Zhang et al., 2023). Let $\boldsymbol{x}$ be an imputation of $\tilde{\boldsymbol{x}}$. The basic idea is that if $\boldsymbol{x}$ is a perfect imputation of $\tilde{\boldsymbol{x}}$, then a classifier cannot predict m_j given $(\boldsymbol{x}, \boldsymbol{m}_{-j})$. For example, given a perfectly imputed image, one cannot tell which pixels are imputed and which pixels are observed. Therefore, in their approach, a generative network is trained to minimize the aforementioned classification accuracy. In fact, this method teaches the generator to *break the dependency between* $\boldsymbol{x}$ and $\boldsymbol{m}$. Inspired by this idea, we propose to impute $\{\tilde{\boldsymbol{x}}\}$ by iteratively updating particles $\{\boldsymbol{x}_t\}$. The initial particle $\boldsymbol{x}_{t=0}$ is set to be

$$x_{0,j} = \begin{cases} \tilde{x}_j & m_j = 1 \\ \text{Sample from e.g., } \mathcal{N}(0,1) & m_j = 0. \end{cases}$$

After that, the particles $\{\boldsymbol{x}_t\}$ are evolved according to the forward KL field that minimizes $\mathrm{KL}[p_{X_t M}, p_{X_t} p_M]$, i.e., the mutual information between X_t (particles) and M (mask). Note that we only update missing dimensions, i.e.,

$$x_{t+1,j} := \begin{cases} x_{t,j} & m_j = 1 \\ x_{t,j} + \eta \partial_{x_j} r_t(\boldsymbol{x}_t, \boldsymbol{m}) & m_j = 0. \end{cases}$$

Samples from $p_{X_t M}$ are available to us since we observe the pairs $\{(\boldsymbol{x}, \boldsymbol{m})\}$ and samples from $p_{X_t} p_M$ can be constructed as $\{(\boldsymbol{x}, \boldsymbol{m}')\}$, where $\boldsymbol{m}'$ is a random sample of M given the missing pattern.

In the first experiment, we test the performance of various imputers on an "S"-shaped dataset, where samples are Missing Completely at Random (MCAR) (Rubin, 1976). The results are plotted in Figure 4. We compare our imputed results (**WGF**) with Optimal Transport-based imputation method (**Sinkhorn**) (Muzellec et al., 2020), and GAN-based imputation method (**GAIN**) (Yoon et al., 2018). σ is chosen by automatic model selection described in Section I.1. It can be seen that our imputer, based on minimizing $\mathrm{KL}[p_{X_t M}, p_{X_t} p_M]$ nicely recovers the "S"-shape after 100 particle update iterations. However, Sinkhorn and GAIN imputation methods struggle to accurately restore the "S"-shaped pattern.

In the second experiment, we test the performance of our algorithm on a real-world Breast Cancer classification dataset (Zwitter and Soklic, 1988) in Figure 5. This is a 30-dimensional binary classification dataset and we artificially create missing values by following the MCAR paradigm with different missing rates. Since the dataset is a binary classification dataset, we compare the performance of linear SVM classifiers trained on imputed datasets. The performance is measured by the Area Under the ROC Curve (AUROC) on hold-out testing sets. In addition to **Sinkhorn** and **GAIN**, we validate the performance of our method with three additional algorithms: MMD flow using negative distance kernel (**MMDFlow**) (Hagemann et al., 2024), a model selection-based method **HyperImpute** (Jarrett et al., 2022), and an auto encoder-based approach (**MIWAE**) (Mattei and Frellsen, 2019). The result shows that SVM trained on the dataset imputed by our method achieves comparable performance to datasets imputed by the other benchmark methods. Our method is robust against the choice of σ. Details and the selection of hyperparameters can be found in Section I.2.

6. Limitations

All f-divergence fields are defined using the density ratio function p/q_t. However, the ratio function is not defined when the input domains of p and q_t are non-overlapping. This problem is particularly noticeable when working on high-dimensional, real-world datasets. This so-called "density chasm problem" (Rhodes et al., 2020) will also affect the velocity field estimation as assumptions required by our estimators e.g., Assumption 4.4 or (13) depends on the density ratio. One possible solution to this issue is to build a "bridge" using interpolations of two distributions, estimate the density ratios of "neighbouring" interpolations, and then combine these estimates. However, this will significantly increase the computational cost as we need to estimate the density ratio gradients of many pairs of distributions. Developing an efficient gradient estimator using target and particle distribution interpolation is an interesting future task.

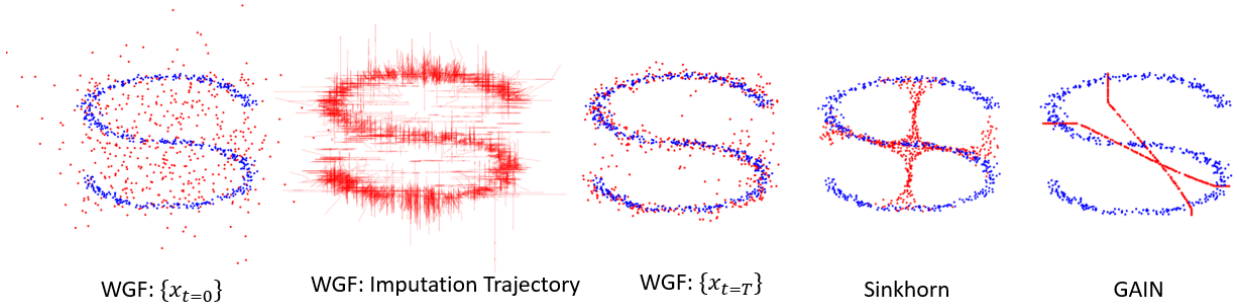


Figure 4. Comparison of imputation methods. Fully observed samples are plotted in blue, and imputed samples in red. The leftmost plot shows the initial particles in the WGF impute. The second left plot visualizes the imputation trajectories of different particles. The third left plot is the final output after 100 WGF iterations.

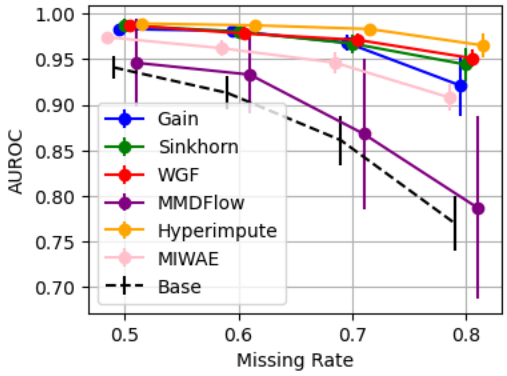


Figure 5. AUROC of a linear SVM classifier on the imputed Breast Cancer dataset. **Base** indicates the performance of a baseline imputer where we impute the missing values with Gaussian noises.

Another potential limitation of the proposed estimator is its applicability to high-dimensional datasets. Although some preliminary investigations show that the proposed estimator does work reasonably well on some high-dimensional generative tasks (see Section O.2), it is known that local regression tends to perform poorly on high-dimensional datasets (see, e.g., (Stone, 1980; 1982)). Instead of performing WGF updates directly on the high-dimensional datasets, we can consider performing the updates in a low-dimensional feature space (see Section L). Some preliminary investigations have shown promising results (see Section O.3).

7. Conclusion

In recent years, it has been discovered that by iteratively updating a set of particles, WGF can approximate a target distribution by reducing the f-divergences between corresponding distributions. This paper addresses the important problem of estimating the velocity fields induced by various f-divergence WGFs. We propose novel interpolation-based estimators for different f-divergences fields and prove their validity. We demonstrate the effectiveness of these estimators through two novel applications: domain adaptation and missing data imputation. Our results show that even without access to the density ratio function, the velocity fields can be efficiently estimated from samples of the target and particle distributions.

Acknowledgments

JS was supported by the EPSRC Centre for Doctoral Training in Computational Statistics and Data Science, grant number EP/S023569/1.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

L. Ambrosio, N. Gigli, and G. Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2005.

A. F. Ansari, M. L. Ang, and H. Soh. Refining deep generative models via discriminator gradient flow. In *International Conference on Learning Representations (ICLR 2021)*, 2021.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

S. Chewi, T. Le Gouic, C. Lu, T. Maunu, and P. Rigollet. Svgd as a kernelized wasserstein gradient flow of the chi-squared divergence. In *Advances in Neural Information*

Processing Systems (NeurIPS 2020), volume 33, pages 2098–2109, 2020.

N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems (NeurIPS 2017)*, volume 30, 2017a.

N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39 (9):1853–1865, 2017b.

M. D. Donsker and S. R. S. Varadhan. Asymptotic evaluation of certain markov process expectations for large time—iii. *Communications on Pure and Applied Mathematics*, 29(4):389–461, 1976.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.

J. Fan. Local Linear Regression Smoothers and Their Minimax Efficiencies. *The Annals of Statistics*, 21(1):196 – 216, 1993.

Y. Gao, Y. Jiao, Y. Wang, Y. Wang, C. Yang, and S. Zhang. Deep generative learning via variational gradient flow. In *International Conference on Machine Learning (ICML 2019)*, pages 2093–2101, 2019.

T. Gasser and H-G Müller. Kernel estimation of regression functions. In T. Gasser and M. Rosenblatt, editors, *Smoothing Techniques for Curve Estimation*, pages 23–68, Berlin, Heidelberg, 1979. Springer Berlin Heidelberg.

J. Givens, S. Liu, and H. W. J. Reeve. Density ratio estimation and neyman pearson classification with missing data. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics (AISTATS 2023)*, volume 206, pages 8645–8681, 2023.

P. Hagemann, J. Hertrich, F. Altekrüger, R. Beinert, J. Chemseddine, and G. Steidl. Posterior sampling based on gradient flows of the MMD with negative distance kernel. In *International Conference on Learning Representations (ICLR 2024)*, 2024.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

A. Hyvärinen. Some extensions of score matching. *Computational Statistics & Data Analysis*, 51(5):2499–2512, 2007.

D. Jarrett, B. Cebere, T. Liu, A. Curth, and M. van der Schaar. Hyperimpute: Generalized iterative imputation with automatic model selection. In *International Conference on Machine Learning (ICML 2022)*, volume 162, page 9916–9937, 2022.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR 2015)*, 2015.

Q. Liu. Stein variational gradient descent as gradient flow. In *Advances in Neural Information Processing Systems (NeurIPS 2017)*, volume 30, pages 3118–3126, 2017.

Q. Liu and D. Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in Neural Information Processing Systems (NeurIPS 2016)*, volume 29, pages 2378–2386, 2016.

D. Maoutsa, S. Reich, and M. Opper. Interacting particle solutions of fokker–planck equations through gradient–log–density estimation. *Entropy*, 22(8):802, 2020.

P-A Mattei and J Frellsen. MIWAE: Deep generative modelling and imputation of incomplete data sets. In *International Conference on Machine Learning (ICML 2019)*, volume 97, pages 4413–4423, 2019.

B. Muzellec, J. Josse, C. Boyer, and M. Cuturi. Missing data imputation using optimal transport. In *International Conference on Machine Learning (ICML 2020)*, pages 7130–7140, 2020.

E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

S. Nowozin, B. Cseke, and R. Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems (NeurIPS 2016)*, volume 29, 2016.

B. Rhodes, K. Xu, and M. U. Gutmann. Telescoping density-ratio estimation. *Advances in Neural Information Processing Systems (NeurIPS 2020)*, 33:4905–4916, 2020.

D. B. Rubin. Inference and Missing Data. *Biometrika*, 63 (3):581–592, 1976. Publisher: Oxford University Press.

D. W. Scott. Feasibility of multivariate density estimates. *Biometrika*, 78(1):197–205, 1991.

J. Simons, S. Liu, and M. Beaumont. Variational likelihood-free gradient descent. In *Fourth Symposium on Advances in Approximate Bayesian Inference (AABI 2021)*, 2021.

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 (UAI 2020)*, pages 574–584, 2020.

C. J. Stone. Optimal rates of convergence for nonparametric estimators. *The Annals of Statistics*, pages 1348–1360, 1980.

C. J. Stone. Optimal Global Rates of Convergence for Nonparametric Regression. *The Annals of Statistics*, 10 (4):1040 – 1053, 1982.

M. Sugiyama, T. Suzuki, and T. Kanamori. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, 2012.

P. Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

M. J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019.

Y. Wang, P. Chen, and W. Li. Projected wasserstein gradient descent for high-dimensional bayesian inference. *SIAM/ASA Journal on Uncertainty Quantification*, 10(4): 1513–1532, 2022.

L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer Publishing Company, Incorporated, 2010.

G. S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 26(4): 359–372, 1964.

A. Wibisono. Sampling as optimization in the space of measures: The langevin dynamics as a composite optimization problem. In *Conference on Learning Theory (COLT 2018)*, pages 2093–3027, 2018.

M. Yi, Z. Zhu, and S. Liu. Monoflow: Rethinking divergence gans via the perspective of wasserstein gradient flows. In *International Conference on Machine Learning (ICML 2023)*, pages 39984–40000, 2023.

J. Yoon, J. Jordon, and M. Schaar. Gain: Missing data imputation using generative adversarial nets. In *International Conference on Machine Learning (ICML 2018)*, pages 5689–5698, 2018.

Q. Yu, A. Hashimoto, and Y. Ushiku. Divergence optimization for noisy universal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021)*, pages 2515–2524, 2021.

Y. Zhang, R. Zhang, and B. Zhao. A systematic review of generative adversarial imputation network in missing data imputation. *Neural Computing and Applications*, 35 (27):19685–19705, September 2023.

M. Zwitter and M. Soklic. Breast Cancer. UCI Machine Learning Repository, 1988.

A. Proof of Proposition 3.1

Proof.

$$
\begin{aligned}
|\hat{u}_i(\boldsymbol{x}^\star) - \partial_i \log r(\boldsymbol{x}^\star)| &= \left| \frac{\widehat{\mathbb{E}}_q\left[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\partial_i \log r(\boldsymbol{x})\right]}{\widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)]} - \partial_i \log r(\boldsymbol{x}^\star) \right| \\
&= \left| \frac{\widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\left(\partial_i \log r(\boldsymbol{x}^\star) + \langle \nabla \partial_i \log r(\bar{\boldsymbol{x}}), \boldsymbol{x} - \boldsymbol{x}^\star \rangle\right)]}{\widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)]} - \partial_i \log r(\boldsymbol{x}^\star) \right| \\
&= \left| \frac{\widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\langle \nabla \partial_i \log r(\bar{\boldsymbol{x}}), \boldsymbol{x} - \boldsymbol{x}^\star \rangle]}{\widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)]} \right| \\
&\leq \frac{\widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\|\nabla \partial_i \log r(\bar{\boldsymbol{x}})\| \cdot \|\boldsymbol{x} - \boldsymbol{x}^\star\|]}{\widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)]} \\
&\leq \kappa \cdot \frac{\widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\|\boldsymbol{x} - \boldsymbol{x}^\star\|]}{\widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)]} \\
&\leq \kappa \cdot \left(\frac{\widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\|\boldsymbol{x} - \boldsymbol{x}^\star\|]}{\widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)]} - \frac{\mathbb{E}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\|\boldsymbol{x} - \boldsymbol{x}^\star\|]}{\mathbb{E}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)]} + \frac{\mathbb{E}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\|\boldsymbol{x} - \boldsymbol{x}^\star\|]}{\mathbb{E}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)]} \right) \qquad (17)
\end{aligned}
$$

The second line is due to the mean value theorem and $\bar{\boldsymbol{x}}$ is a point in between $\boldsymbol{x}$ and $\boldsymbol{x}^\star$ in a coordinate-wise fashion. The second inequality is due to the operator norm of a matrix is always greater than a row/column norm.

Since $\mathrm{Var}_q[\frac{1}{\sigma^d} k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\|\boldsymbol{x} - \boldsymbol{x}^\star\|] = O(\frac{1}{\sigma^d})$ due to the assumption, using Chebyshev inequality

$$
\widehat{\mathbb{E}}_q\left[\frac{1}{\sigma^d} k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\|\boldsymbol{x} - \boldsymbol{x}^\star\|\right] - \mathbb{E}_q\left[\frac{1}{\sigma^d} k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\|\boldsymbol{x} - \boldsymbol{x}^\star\|\right] = O_p(\frac{1}{\sqrt{n\sigma^d}}).
$$

Similarly, due to the assumption that $\mathrm{Var}_q[\frac{1}{\sigma^d} k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)] = O(\frac{1}{\sigma^d})$, we have

$$
\widehat{\mathbb{E}}_q\left[\frac{1}{\sigma^d} k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\right] - \mathbb{E}_q\left[\frac{1}{\sigma^d} k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\right] = O_p(\frac{1}{\sqrt{n\sigma^d}}).
$$

Lemma A.1. *Suppose* $|\mathbb{E}A| \leq A_1 < \infty, |\mathbb{E}B| > B_0 > 0$. $\widehat{\mathbb{E}}A - \mathbb{E}A = O_p(\frac{1}{\sqrt{n\sigma^d}})$ *and* $\widehat{\mathbb{E}}B - \mathbb{E}B = O_p(\frac{1}{\sqrt{n\sigma^d}})$

$$
\frac{\widehat{\mathbb{E}}A}{\widehat{\mathbb{E}}B} - \frac{\mathbb{E}A}{\mathbb{E}B} = O_p\left(\frac{1}{\sqrt{n\sigma^d}}\right).
$$

Sketch of proof

The argument below resembles the proof technique used in A.2 in Givens et al. (2023). Here, we provide a sketch argument.

$\left|\frac{\widehat{\mathbb{E}}A}{\widehat{\mathbb{E}}B} - \frac{\mathbb{E}A}{\mathbb{E}B}\right| = \left|\frac{\widehat{\mathbb{E}}A\mathbb{E}B - \mathbb{E}A\widehat{\mathbb{E}}B}{\widehat{\mathbb{E}}B\mathbb{E}B}\right| \leq \left|\frac{\widehat{\mathbb{E}}A - \mathbb{E}A}{\widehat{\mathbb{E}}B}\right| + \left|\frac{\widehat{\mathbb{E}}B - \mathbb{E}B}{\widehat{\mathbb{E}}B\mathbb{E}B}\right| \cdot |\mathbb{E}A|$.

First, due to the boundedness of $|\mathbb{E}B|$, for any ϵ, $\exists N, \forall n > N$, $|\widehat{\mathbb{E}}B| > |\mathbb{E}B/2| > B_0/2$ with a probability $1 - \epsilon$. Thus, $\widehat{\mathbb{E}}A - \mathbb{E}A = O_p(\frac{1}{\sqrt{n\sigma^d}})$ implies $\left|\frac{\widehat{\mathbb{E}}A - \mathbb{E}A}{\widehat{\mathbb{E}}B}\right| = O_p(\frac{1}{\sqrt{n\sigma^d}})$.

Second, due to the boundedness of $|\mathbb{E}A|$, and the boundedness of $|\widehat{\mathbb{E}}B|$ argued above, $\widehat{\mathbb{E}}A - \mathbb{E}A = O_p(\frac{1}{\sqrt{n\sigma^d}})$ implies $\left|\frac{\widehat{\mathbb{E}}B - \mathbb{E}B}{\widehat{\mathbb{E}}B\mathbb{E}B}\right| \cdot |\mathbb{E}A| = O_p(\frac{1}{\sqrt{n\sigma^d}})$.

Hence, $\frac{\widehat{\mathbb{E}}A}{\widehat{\mathbb{E}}B} - \frac{\mathbb{E}A}{\mathbb{E}B} = O_p\left(\frac{1}{\sqrt{n\sigma^d}}\right)$.

Due to Lemma A.1 and (17), we have with high probability that

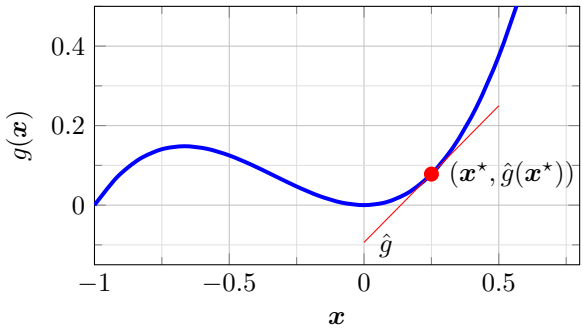


Figure 6. Fit g locally at $\boldsymbol{x}^\star$ using a linear function $\hat{g}$. Its "slope" is a natural estimator of $\nabla_{\boldsymbol{x}} g(\boldsymbol{x}^\star)$.

$$
\begin{aligned}
|\hat{u}_i(\boldsymbol{x}^\star) - \partial_i \log r(\boldsymbol{x}^\star)| \leq & \kappa \cdot \left(\frac{K}{\sqrt{n\sigma^d}} + \frac{\mathbb{E}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\|\boldsymbol{x} - \boldsymbol{x}^\star\|]}{\mathbb{E}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)]} \right) \\
\leq & \kappa \cdot \left(\frac{K}{\sqrt{n\sigma^d}} + \frac{\sigma \int q(\boldsymbol{x}^\star + \sigma \boldsymbol{y}) k(\boldsymbol{y}) \|\boldsymbol{y}\| \mathrm{d}\boldsymbol{y}}{\int q(\boldsymbol{x}^\star + \sigma \boldsymbol{y}) k(\boldsymbol{y}) \mathrm{d}\boldsymbol{y}} \right) \qquad \boldsymbol{y} := (\boldsymbol{x} - \boldsymbol{x}^\star)/\sigma \\
\leq & \kappa \cdot \left(\frac{K}{\sqrt{n\sigma^d}} + \sigma C_k \right),
\end{aligned}
$$

where K is constant.

□

B. Visualization of Gradient Estimation using Local Linear Fitting

C. Variational Objective for Estimating $h \circ r$

Proposition C.1. *The maximum in* (9) *is attained if and only if* $d = h \circ r$.

Proof. Due to the maximizing argument, the maximum of (9) is attained if and only if $d = \psi'$. The definition of mirror divergence indicates that $\psi' = r\phi'(r) - \phi(r)$. Theorem 2.1 states that $h \circ r = r\phi'(r) - \phi(r) = \psi'$, thus the maximum is attained if and only if $d = h \circ r$. □

D. Proof of Theorem 4.8, Estimation Error Bound of $\boldsymbol{w}(\boldsymbol{x}^\star)$

Proof. In this section, to simplify notations, we denote $\boldsymbol{\theta}$ as the parameter vector that combines both $\boldsymbol{w}$ and b, i.e., $\boldsymbol{\theta} := [\boldsymbol{w}, b]^\top$. Specifically, we define

$$
\boldsymbol{\theta}^* = \left[\boldsymbol{w}^{*\top}, b^* \right]^\top := \left[\nabla^\top (h \circ r)(\boldsymbol{x}^\star), h(r(\boldsymbol{x}^\star)) - \langle \nabla (h \circ r)(\boldsymbol{x}^\star), \boldsymbol{x}^\star \rangle \right]^\top .
$$

Let us denote the negative objective function in (10) as $\ell(\boldsymbol{\theta})$ and consider a constrained optimization problem:

$$
\min_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) \text{ subject to:} \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2 \leq \min(W, B)^2. \tag{18}
$$

This convex optimization has a Lagrangian $\ell(\boldsymbol{\theta}) + \lambda(\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2 - \min(W, B)^2)$, where $\lambda \geq 0$ is the Lagrangian multiplier. According to KKT condition, the optimal solution $\boldsymbol{\theta}_0$ of (18) satisfies $\nabla \ell(\boldsymbol{\theta}_0) + 2\lambda(\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*) = \mathbf{0}$.

We apply mean value theorem to $g(\boldsymbol{\theta}) := \langle \boldsymbol{\theta}_0 - \boldsymbol{\theta}^*, \nabla \ell(\boldsymbol{\theta}) + 2\lambda(\boldsymbol{\theta} - \boldsymbol{\theta}^*) \rangle$:

$$
\underbrace{\langle \boldsymbol{\theta}_0 - \boldsymbol{\theta}^*, \nabla \ell(\boldsymbol{\theta}_0) + 2\lambda(\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*) \rangle}_{g(\boldsymbol{\theta}_0)=0, \text{ KKT condition}} = \underbrace{\langle \boldsymbol{\theta}_0 - \boldsymbol{\theta}^*, \nabla \ell(\boldsymbol{\theta}^*) \rangle}_{g(\boldsymbol{\theta}^*)} + \langle \underbrace{(\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*)^\top \left[\nabla^2 \ell(\bar{\boldsymbol{\theta}}) + 2\lambda \boldsymbol{I} \right]}_{\nabla g(\bar{\boldsymbol{\theta}})}, (\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*) \rangle,
$$

where $\bar{\boldsymbol{\theta}}$ is a point between $\boldsymbol{\theta}^*$ and $\boldsymbol{\theta}_0$ in an elementwise fashion. Since $\bar{\boldsymbol{\theta}}$ is in a hyper cube with $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}^*$ as opposite corners, it is in the constrain set of (18). Let us rearrange terms:

$$-\langle \boldsymbol{\theta}_0 - \boldsymbol{\theta}^*, \nabla \ell(\boldsymbol{\theta}^*)\rangle = \langle \boldsymbol{\theta}_0 - \boldsymbol{\theta}^*, \left[\nabla^2 \ell(\bar{\boldsymbol{\theta}}) + 2\lambda \boldsymbol{I}\right](\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*)\rangle \geq \langle \boldsymbol{\theta}_0 - \boldsymbol{\theta}^*, \nabla^2 \ell(\bar{\boldsymbol{\theta}})(\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*)\rangle,$$

For all $\boldsymbol{\theta}$ in the constraint set, $\|\boldsymbol{w}\| = \|\boldsymbol{w} - \boldsymbol{w}^* + \boldsymbol{w}^*\| \leq \|\boldsymbol{w} - \boldsymbol{w}^*\| + \|\boldsymbol{w}^*\| \leq \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| + \|\boldsymbol{w}^*\| \leq 2W$ and $|b| = |b - b^* + b^*| \leq |b - b^*| + |b^*| \leq \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| + |b^*| \leq 2B$. Under our assumption (14), the lowest eigenvalue of $\nabla^2 \ell(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$ in the constrain set of (18) is always lower bounded by $\sigma^d \Lambda_{\min}$. Therefore,

$$-\langle \boldsymbol{\theta}_0 - \boldsymbol{\theta}^*, \nabla \ell(\boldsymbol{\theta}^*)\rangle \geq \sigma^d \cdot \Lambda_{\min} \|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*\|^2.$$

Using Cauchy–Schwarz inequality

$$\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*\| \|\nabla \ell(\boldsymbol{\theta}^*)\| \geq \sigma^d \cdot \Lambda_{\min} \|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*\|^2.$$

Assume $\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*\|$ is not zero (if it is, our estimator is already consistent).

$$\begin{aligned}
\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*\| \leq & \frac{1}{\sigma^d \Lambda_{\min}} \|\nabla \ell(\boldsymbol{\theta}^*)\| \\
\leq & \frac{1}{\sigma^d \Lambda_{\min}} \|\nabla \ell(\boldsymbol{\theta}^*) - \mathbb{E}\nabla \ell(\boldsymbol{\theta}^*) + \mathbb{E}\nabla \ell(\boldsymbol{\theta}^*)\| \\
\leq & \frac{1}{\sigma^d \Lambda_{\min}} \left(\|\nabla \ell(\boldsymbol{\theta}^*) - \mathbb{E}\nabla \ell(\boldsymbol{\theta}^*)\| + \|\mathbb{E}\nabla \ell(\boldsymbol{\theta}^*)\|\right)
\end{aligned} \tag{19}$$

Since $\operatorname{tr}\left[\operatorname{Cov}_p\left[\frac{1}{\sigma^d} k(\boldsymbol{x}, \boldsymbol{x}^\star) \cdot \tilde{\boldsymbol{x}}\right]\right] = O(\frac{1}{\sigma^d})$ and $\operatorname{tr}\left[\operatorname{Cov}_q\left[\frac{1}{\sigma^d} k(\boldsymbol{x}, \boldsymbol{x}^\star) \cdot \psi_{\text{con}}(\langle \boldsymbol{w}^*, \boldsymbol{x}\rangle + b^*)\tilde{\boldsymbol{x}}\right]\right] = O(\frac{1}{\sigma^d})$ by assumption, due to multi-dimensional version of Chebyshev's inequality, $\sigma^d \| \frac{1}{\sigma^d} \left(\nabla \ell(\boldsymbol{\theta}^*) - \mathbb{E}\nabla \ell(\boldsymbol{x}^*)\right) \| = \sigma^d O_p(\frac{1}{\sqrt{n\sigma^d}})$. Therefore,

$$\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*\| \leq \frac{1}{\sigma^d \Lambda_{\min}} \left(\sigma^d \cdot \frac{K}{\sqrt{n\sigma^d}} + \|\mathbb{E}\nabla \ell(\boldsymbol{\theta}^*)\|\right),$$

with high probability and K is a constant.

Now we proceed to bound $\|\mathbb{E}\nabla \ell(\boldsymbol{\theta}^*)\|$.

Lemma D.1. $\|\mathbb{E}\nabla \ell(\boldsymbol{\theta}^*)\| \leq \mathbb{E}_q\left[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star) \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{x}^\star\|^2 \|\tilde{\boldsymbol{x}}\|\right] \kappa \cdot C_{\psi''_{\text{con}}}$

Proof. The expression of $\mathbb{E}\nabla \ell(\boldsymbol{\theta}^*)$ is:

$$\mathbb{E}\nabla \ell(\boldsymbol{\theta}^*) := -\mathbb{E}_p\left[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star) \cdot \tilde{\boldsymbol{x}}\right] + \mathbb{E}_q\left[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star) \psi'_{\text{con}}(\langle \boldsymbol{w}^*, \boldsymbol{x}\rangle + b^*) \cdot \tilde{\boldsymbol{x}}\right]. \tag{20}$$

Due to Taylor's theorem, $\langle \boldsymbol{w}^*, \boldsymbol{x}\rangle + b^* = h(r(\boldsymbol{x})) - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^\star)^\top \nabla^2 (h \circ r)(\bar{\boldsymbol{x}})(\boldsymbol{x} - \boldsymbol{x}^\star)$ where $\bar{\boldsymbol{x}}$ is a point in between $\boldsymbol{x}$ and $\boldsymbol{x}^\star$ in an elementwise fashion. Thus, applying the mean value theorem on ψ'_{con},

$$\begin{aligned}
\psi'_{\text{con}}(\langle \boldsymbol{w}^*, \boldsymbol{x}\rangle + b^*) &= \psi'_{\text{con}}\left[h(r(\boldsymbol{x})) - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^\star)^\top \nabla^2 (h \circ r)(\bar{\boldsymbol{x}})(\boldsymbol{x} - \boldsymbol{x}^\star)\right] \\
&= \psi'_{\text{con}}\left[h(r(\boldsymbol{x}))\right] - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^\star)^\top \nabla^2 (h \circ r)(\bar{\boldsymbol{x}})(\boldsymbol{x} - \boldsymbol{x}^\star) \psi''_{\text{con}}(y),
\end{aligned}$$

where y is a scalar in between $h(r(\boldsymbol{x}))$ and $h(r(\boldsymbol{x})) - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^\star)^\top \nabla^2 (h \circ r)(\bar{\boldsymbol{x}})(\boldsymbol{x} - \boldsymbol{x}^\star)$ or equivalently, in between $h(r(\boldsymbol{x}))$ and $\langle \boldsymbol{w}^*, \boldsymbol{x}\rangle + b^*$.

Theorem 2.1 states that $h(r) = r\phi' + \phi$, then, by the definition of the mirror divergence, $\psi' = h(r)$. Moreover, due to the maximizing argument, ψ'_{con} is the input argument of ψ (i.e., r) and ψ' is the input argument of ψ_{con}. Thus, $\psi'_{\text{con}}(h(r)) = \psi'_{\text{con}}(\psi'(r)) = r$. Let us write

$$\begin{aligned}
\mathbb{E}\nabla \ell(\boldsymbol{\theta}^*) = & -\mathbb{E}_p[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star)\tilde{\boldsymbol{x}}] + \mathbb{E}_q[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star) \underbrace{\psi'_{\text{con}}(h(r(\boldsymbol{x})))}_{r} \tilde{\boldsymbol{x}}] \\
& - \mathbb{E}_q\left[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star) \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^\star)^\top \nabla^2 (h \circ r)(\bar{\boldsymbol{x}})(\boldsymbol{x} - \boldsymbol{x}^\star) \psi''_{\text{con}}(y) \tilde{\boldsymbol{x}}\right] \\
= & -\mathbb{E}_q\left[k_\sigma(\boldsymbol{x}, \boldsymbol{x}^\star) \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^\star)^\top \nabla^2 (h \circ r)(\bar{\boldsymbol{x}})(\boldsymbol{x} - \boldsymbol{x}^\star) \psi''_{\text{con}}(y) \tilde{\boldsymbol{x}}\right],
\end{aligned}$$

We can derive a bound for $\|\mathbb{E}\nabla\ell(\boldsymbol{\theta}^*)\|$:

$$\begin{aligned}
\|\mathbb{E}\nabla\ell(\boldsymbol{\theta}^*)\| &\leq \mathbb{E}_q\left[k_\sigma(\boldsymbol{x},\boldsymbol{x}^\star)\cdot\frac{1}{2}\left|(\boldsymbol{x}-\boldsymbol{x}^\star)^\top\nabla^2(h\circ r)(\bar{\boldsymbol{x}})\,(\boldsymbol{x}-\boldsymbol{x}^\star)\right|\cdot|\psi''_{\mathrm{con}}(y)|\cdot\|\tilde{\boldsymbol{x}}\|\right]\\
&\leq \mathbb{E}_q\left[k_\sigma(\boldsymbol{x},\boldsymbol{x}^\star)\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{x}^\star\|^2\left\|\nabla^2(h\circ r)(\bar{\boldsymbol{x}})\right\|\|\tilde{\boldsymbol{x}}\|\right]C_{\psi''_{\mathrm{con}}}\\
&\leq \mathbb{E}_q\left[k_\sigma(\boldsymbol{x},\boldsymbol{x}^\star)\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{x}^\star\|^2\|\tilde{\boldsymbol{x}}\|\right]\kappa\cdot C_{\psi''_{\mathrm{con}}}\\
&\leq \sigma^{d+2}\cdot\kappa\cdot C_{\psi''_{\mathrm{con}}}\frac{1}{2}\int q(\sigma\boldsymbol{y}+\boldsymbol{x}^\star)\cdot k(\boldsymbol{y})\cdot\|\boldsymbol{y}\|^2\cdot\|[\sigma\boldsymbol{y}+\boldsymbol{x}^\star,1]\|\,\mathrm{d}\boldsymbol{y}\leq\sigma^{d+2}\cdot\kappa\cdot C_{\psi''_{\mathrm{con}}}C_k
\end{aligned}$$

□

Finally, due to Lemma D.1 and Assumption 4.5, we can see that with high probability

$$\|\boldsymbol{\theta}^*-\boldsymbol{\theta}_0\|\leq\frac{\frac{K\sigma^d}{\sqrt{n\sigma^d}}+\kappa\cdot\sigma^{d+2}\cdot C_k\cdot C_{\psi''_{\mathrm{con}}}}{\sigma^d\Lambda_{\min}}\leq\frac{\frac{K}{\sqrt{n\sigma^d}}+\kappa\cdot\sigma^2\cdot C_k\cdot C_{\psi''_{\mathrm{con}}}}{\Lambda_{\min}}.$$

Since $n\sigma^d\to\infty,\sigma\to 0$, $\|\boldsymbol{\theta}^*-\boldsymbol{\theta}_0\|\to 0$ with high probability. There always exists σ_0 and N, such that for all $\sigma<\sigma_0$ and $n>N$, $\min(W,B)>\frac{\frac{K}{\sqrt{n\sigma^d}}+\kappa\cdot C_k\cdot C_{\psi''_{\mathrm{con}}}\cdot\sigma^2}{\Lambda_{\min}}$. When it happens, $\boldsymbol{\theta}_0$ must be the interior of the constrain set of (18). i.e., the constraints in (18) are not active. It implies $\boldsymbol{\theta}_0$ must be the stationary point of $\ell(\boldsymbol{\theta})$ as long as σ is sufficiently small and n is sufficiently large.

□

E. Proof of Corollary 4.9

Proof. Since (11) has a unconstrained quadratic objective, its maximizers are stationary points.

We can see that Assumption 4.4 holds. To apply Theorem 4.8, we still need to show that Assumption 4.7 holds and W and B exist. In this case, $\psi_{\mathrm{con}}(d)=d^2/2+d$, so $\psi''_{\mathrm{con}}=1$. Thus Assumption 4.7 holds automatically for every $C_{\psi''_{\mathrm{con}}}\geq 1$. Additionally,

$$\nabla^2_{[\boldsymbol{w},b]}\psi_{\mathrm{con}}(\langle\boldsymbol{w},\boldsymbol{x}\rangle+b)=\widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x},\boldsymbol{x}^\star)\psi''_{\mathrm{con}}(\langle\boldsymbol{w},\boldsymbol{x}\rangle+b)\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\top]=\widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x},\boldsymbol{x}^\star)\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\top].$$

Therefore, (15) implies the minimum eigenvalue assumption (14) holds for every $W>0,B>0$. Thus, we can choose any $W>0$ and $B>0$ that satisfies (13). Noticing that $h(r(\boldsymbol{x}))=r(\boldsymbol{x})$, applying Theorem 4.8 gives the desired result. □

F. Proof of Corollary 4.10

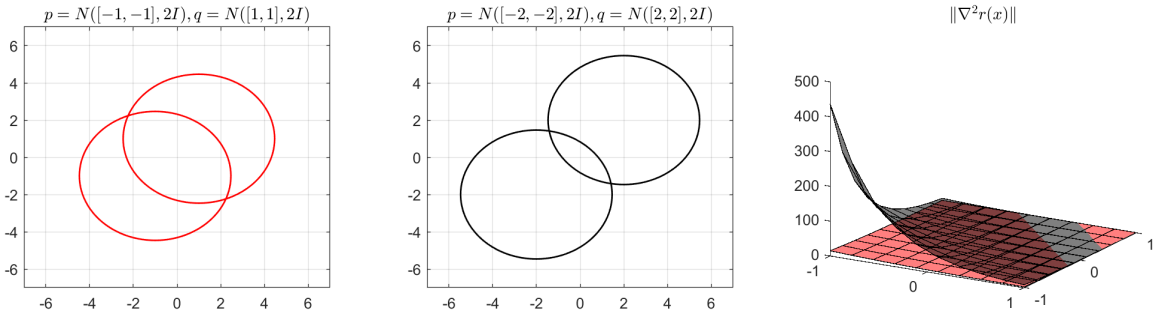
Proof. Assumption 4.4, 4.5 and (13) are already satisfied. Let us verify the eigenvalue condition (14). In this case, $\psi''_{\mathrm{con}}(d)=\exp(d-1)$. Thus $\exp(\langle\boldsymbol{w},\boldsymbol{x}\rangle+b-1)\leq\exp(2WC_{\mathcal{X}}+2B-1)<\infty$ for $\|\boldsymbol{w}\|\leq 2W,|b|\leq 2B$. Moreover, because $\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\top$ is positive semi-definite, due to (16),

$$\begin{aligned}
\lambda_{\min}\left[\widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x},\boldsymbol{x}^\star)\exp(\langle\boldsymbol{w},\boldsymbol{x}\rangle+b-1)\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\top]\right]\geq&\sigma^2\cdot\lambda_{\min}\left[\widehat{\mathbb{E}}_q[k_\sigma(\boldsymbol{x},\boldsymbol{x}^\star)\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\top]\right]\cdot\exp(-2WC_{\mathcal{X}}-2B-1)\\
>&\sigma^2\cdot\Lambda_{\leftarrow}\cdot\exp(-2WC_{\mathcal{X}}-2B-1)>0,
\end{aligned}$$

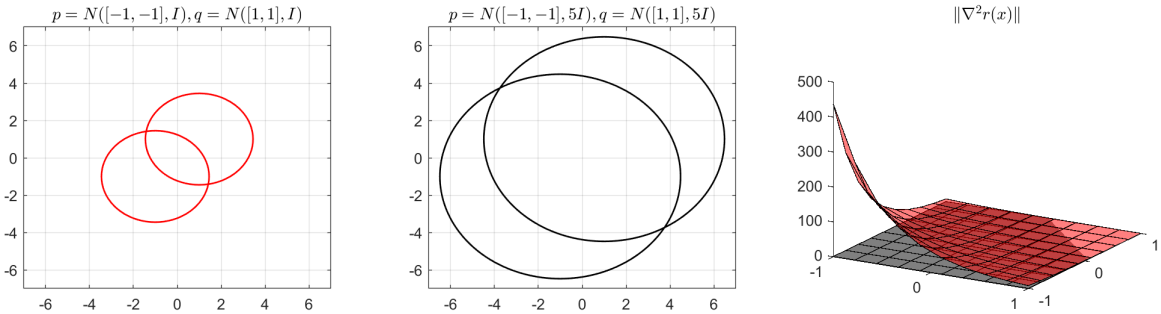
for all $\boldsymbol{w},b$ that $\|\boldsymbol{w}\|\leq 2W,|b|\leq 2B$. So (14) holds. Finally, let us verify Assumption 4.7. Since $\psi_{\mathrm{con}''}$ is a strictly monotone increasing function, $\sup_{a\in[0,1]}\psi_{\mathrm{con}''}(ax_0+(1-a)y_0)$ is obtained either at x_0 or y_0. We only need to verify that $\psi_{\mathrm{con}''}(h(r(\boldsymbol{x})))$ and $\psi_{\mathrm{con}''}(\langle\boldsymbol{w}^*,\boldsymbol{x}\rangle+b^*)$ are both bounded for all $\boldsymbol{x}$. Both $h(r(\boldsymbol{x}))$ and $\langle\boldsymbol{w}^*,\boldsymbol{x}\rangle+b^*$ can be bounded using our assumptions. Thus, for a

$$C_{\psi''_{\mathrm{con}}}=\exp(W\cdot C_{\mathcal{X}}+B-1)\vee\exp(C_{\log r}-1)$$

Assumption 4.7 holds. Applying Theorem 4.8 completes the proof. □



(a) p, q with different means. The further p, q are apart, the larger the $\sup_{\boldsymbol{x}} \|\nabla^2 r(\boldsymbol{x})\|$ is.



(b) p, q with different variances. The less overlapped p, q are, the larger the $\sup_{\boldsymbol{x}} \|\nabla^2 r(\boldsymbol{x})\|$ is.

Figure 7. Visualizing $\nabla^2(h \circ r)(\boldsymbol{x})$ for $\mathrm{D}_\phi = \mathrm{KL}[p, q]$ with two different settings of p and q.

G. $\|\nabla^2 r(\boldsymbol{x})\|$ for Different p and q

See Figure 7.

H. Finite-sample Objectives

H.1. Practical Implementation

In our experiments, we observe that (10) can be efficiently minimized by using gradient descent with adaptive learning rate schemes, e.g., Adam (Kingma and Ba, 2015). One computational advantage of the local linear model is that the computation for each $\boldsymbol{x}^\star$ is independent from the others. This property allows us to parallelize the optimization. Even using a single CPU/GPU, we can easily write highly vectorized code to compute the gradient of (10) with respect to $\boldsymbol{w}$ and b for a large particle set $\{\boldsymbol{x}_i^\star\}_{i=1}^n$.

Suppose $\boldsymbol{X}_p \in \mathbb{R}^{n_p \times d}$ and $\boldsymbol{X}_q \in \mathbb{R}^{n_q \times d}$ are the matrices whose rows are $\boldsymbol{x}_p^{(i)}$ and $\boldsymbol{x}_q^{(i)}$ respectively. $\boldsymbol{K}_p \in \mathbb{R}^{n \times n_p}$, $\boldsymbol{K}_q \in \mathbb{R}^{n \times n_q}$ are the kernel matrices between $\{\boldsymbol{x}^\star\}$ and $\boldsymbol{X}_p$, $\{\boldsymbol{x}^\star\}$ and $\boldsymbol{X}_q$ respectively. $\boldsymbol{W} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{b} \in \mathbb{R}^n$ are the parameters whose rows are $\boldsymbol{w}(\boldsymbol{x}^\star)$ and $b(\boldsymbol{x}^\star)$ respectively. Then the gradient of (10) with respect to $\boldsymbol{W}$, $\boldsymbol{b}$ can be expressed as

$$\boldsymbol{K}_p \boldsymbol{X}_p / n_p - \boldsymbol{K}_q \odot \psi'_{\text{con}}(\boldsymbol{W}\boldsymbol{X}_q^\top + \boldsymbol{b})\boldsymbol{X}_q / n_q,$$
$$\boldsymbol{K}_p \mathbf{1}_{n_p} / n_p - \boldsymbol{K}_q \odot \psi'_{\text{con}}(\boldsymbol{W}\boldsymbol{X}_q^\top + \boldsymbol{b})\mathbf{1}_{n_q} / n_q,$$

where $\mathbf{1}_{n_p}$ and $\mathbf{1}_{n_q}$ are the vectors of ones with length n_p and n_q respectively. ψ'_{con} is evaluated element-wise. $\odot$ is the element-wise product and the vector $\boldsymbol{b}$ is broadcast to a matrix with n_q columns.

I. Experiment Details in Section 5

I.1. Model Selection

Let $\mathcal{D}_p := \left\{\boldsymbol{x}_p^{(i)}\right\}_{i=1}^{n_p}, \mathcal{D}_q := \left\{\boldsymbol{x}_q^{(i)}\right\}_{i=1}^{n_q}$ be training sets from p and q respectively and $\tilde{\mathcal{D}}_p = \left\{\tilde{\boldsymbol{x}}_p^{(i)}\right\}_{i=1}^{\tilde{n}_p}$ and $\tilde{\mathcal{D}}_q = \left\{\tilde{\boldsymbol{x}}_q^{(i)}\right\}_{i=1}^{\tilde{n}_q}$ be testing sets. We can fit a local linear model *at each testing point using the training sets*, i.e.,

$$\left(\hat{\boldsymbol{w}}_\sigma(\tilde{\boldsymbol{x}}), \hat{b}_\sigma(\tilde{\boldsymbol{x}})\right) := \underset{\boldsymbol{w}\in\mathbb{R}^d, b\in\mathbb{R}}{\operatorname{argmax}} \underbrace{\ell\left(\boldsymbol{w}, b; \tilde{\boldsymbol{x}}, \mathcal{D}_p, \mathcal{D}_q\right)}_{\text{tranining objective}}.$$

The dependency on σ comes from the smoothing kernel in the training objective. We can tune σ by evaluating the variational lower bound (9) approximated using testing samples:

$$\tilde{\ell}\left(\sigma; \tilde{\mathcal{D}}_p, \tilde{\mathcal{D}}_q\right) := \underbrace{\tilde{\mathbb{E}}_p\left[\hat{d}_\sigma\left(\boldsymbol{x}\right)\right] - \tilde{\mathbb{E}}_q\left[\psi_{\text{con}}(\hat{d}_\sigma\left(\boldsymbol{x}\right))\right]}_{\text{testing criterion}}, \tag{21}$$

where $\tilde{\mathbb{E}}_p[\hat{d}(\boldsymbol{x})] := \frac{1}{\tilde{n}_p}\sum_{i=1}^{\tilde{n}_p}\hat{d}(\tilde{\boldsymbol{x}}_p^{(i)})$, i.e., the sample average over the testing points. $\hat{d}_\sigma(\tilde{\boldsymbol{x}}) := \langle\hat{\boldsymbol{w}}_\sigma\left(\tilde{\boldsymbol{x}}\right), \tilde{\boldsymbol{x}}\rangle + \hat{b}_\sigma\left(\tilde{\boldsymbol{x}}\right)$, is the interpolation of $d(\tilde{\boldsymbol{x}})$ using training samples. The best choice of σ should maximize the above testing criterion.

In our experiments, we construct training and testing sets using cross validation and choose a list of candidate σ for the model selection. This procedure is parallel to selecting k in k-nearest neighbors to minimize the testing error. In our case, (21) is the "negative testing error".

I.2. Missing Data Imputation

Before running the experiments, we first pre-process data in the following way:

1. Suppose $\boldsymbol{X}_{\text{true}}$ is the original data matrix, i.e. without missing values. We introduce missingness to $\boldsymbol{X}_{\text{true}}$, and call the matrix with missing values $\tilde{\boldsymbol{X}}$, following MCAR paradigm. Denote the corresponding mask matrix as $\boldsymbol{M}$, where $m_j^{(i)} = 0$ if $\tilde{x}_j^{(i)}$ is missing, and $m_j^{(i)} = 1$ otherwise.

2. Calculate column-wise mean $\bar{\tilde{\boldsymbol{x}}}$ and standard deviation $\tilde{\boldsymbol{s}}$ (excluding missing values) of $\tilde{\boldsymbol{X}}$.

3. Standardize $\tilde{\boldsymbol{X}}$ by taking $\tilde{\boldsymbol{X}} = \frac{\tilde{\boldsymbol{X}} - \bar{\tilde{\boldsymbol{x}}}}{\tilde{\boldsymbol{s}}}$, where the vectors $\bar{\tilde{\boldsymbol{x}}}$ and $\tilde{\boldsymbol{s}}$ are broadcasted to the same dimensions as the matrix $\tilde{\boldsymbol{X}}$. Note that the division here is element-wise.

Denote $\boldsymbol{X}^t$ as the imputed data of $\tilde{\boldsymbol{X}}$ at iteration i, where $\boldsymbol{X}^0 = \tilde{\boldsymbol{X}} \odot \boldsymbol{M} + \mathbf{Z} \odot (1 - \boldsymbol{M})$, and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \operatorname{diag}(\mathbf{1}))$.

We performed two experiments on both toy data ("S"-shape) and real world data (UCI Breast Cancer [3] data).

Let N_{WGF} be the number of iterations WGF is performed. In each iteration, let N_{GradEst} be the number gradient descent steps for gradient estimation. In the missing data experiments, we set the hyperparameters to be:

- "S"-shape data: $T_{\text{WGF}} = 100$, $T_{\text{GradEst}} = 2000$, σ is chosen by model selection described in Section I.1.
- UCI Breast Cancer data: $T_{\text{WGF}} = 1000$, $T_{\text{GradEst}} = 100$, $\sigma = \text{median}(\sqrt{\frac{\text{pairwise distance of}\boldsymbol{X}}{2}})$.

Across experiments, we observe that our method is robust against the choice of the bandwidth σ. We demonstrate this via a new experiment that tests the imputation performance with different selections of σ. Figure 8 demonstrates this experiment on the UCI Breast Cancer data. We can see that the imputation performance is still comparable with the original results (and baseline methods) when the scale of σ varies from 0.01 to 1.

[3] Available at https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.data.

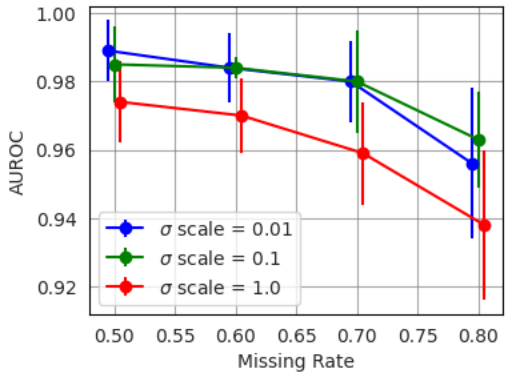


Figure 8. AUROC of a linear SVM classifier on the imputed Breast Cancer dataset, with various scales of σ from 0.01 to 1.

I.3. Wasserstein Gradient Flow

In this experiment, we first expand MNIST digits into 32×32 pictures then adds a small random noise $\epsilon \sim \mathcal{N}(\mathbf{0}, 0.001^2 \boldsymbol{I})$ to each picture so that computing the sample mean and covariance will not cause numerical issues. For both forward and backward KL WGF, we use a kernel bandwidth that equals to $1/5$ of the pairwise distances in the particle dataset, as it is too computationally expensive to perform cross validation at each iteration. After each update, we clip pixel values so that they are in between $[0, 1]$. It is done using pyTorch `torch.clamp` function.

To reduce computational cost, at each iteration, we randomly select 4000 samples from the original dataset and 4000 particles from the particle set. We use these samples to estimate the WGF updates.

J. Discussion:Kernel Density Gradient Estimation

The Kernel Density Estimator (KDE) of p is

$$\hat{p}(\boldsymbol{y}) := \frac{1}{n_p} \sum_{i=1}^{n_p} k(\boldsymbol{x}_p^{(i)}, \boldsymbol{y})/Z,$$

where Z is a normalization constant to ensure that $\int \hat{p}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 1$. Thus,

$$\nabla_{\boldsymbol{y}} \log \hat{p}(\boldsymbol{y}) := \frac{1}{n_p} \sum_{i=1}^{n_p} \nabla_{\boldsymbol{y}} k(\boldsymbol{x}_p^{(i)}, \boldsymbol{y}) / \frac{1}{n_p} \sum_{i=1}^{n_p} k(\boldsymbol{x}_p^{(i)}, \boldsymbol{y}).$$

The normalizing constant Z is cancelled.

K. Discussion: Why Score Matching does not Work on Log Ratio Gradient Estimation

For Score Matching (SM), the estimator of $\hat{p} := \operatorname*{argmin}_f \int p \|\nabla \log p - \nabla \log f\|^2 \mathrm{d}\boldsymbol{x}$, where the objective function is commonly refered to as *Fisher Divergence*. To use SM in practice, the objective function is further broken down to

$$\int p \|\nabla \log p - \nabla \log f\|^2 \mathrm{d}\boldsymbol{x} = \int p \|\nabla \log f\|^2 \mathrm{d}\boldsymbol{x} + 2 \sum_{i=1}^{d} \int p \partial_i^2 \log f \mathrm{d}\boldsymbol{x} + C, \tag{22}$$

where we used the dimension-wise integration by parts and C is a constant.

Since our target is to estimate $\nabla \log p$, we can directly model $\nabla \log p$ as $\boldsymbol{g} : \mathbb{R}^d \to \mathbb{R}^d$. The objective becomes

$$\int p \|\boldsymbol{g}\|^2 \mathrm{d}\boldsymbol{x} + 2 \sum_{i=1}^{d} \int p \partial_i g_i \mathrm{d}\boldsymbol{x} + C.$$

SM can be used to estimate $\nabla \log p$. One might assume that SM can also be used for estimating $\nabla \log r$, where $r := \frac{p}{q}$. Let us replace $\nabla \log p$ with $\nabla \log r$ in (22),

$$\begin{aligned}
\int p\|\nabla \log r - \nabla \log f\|^2 \mathrm{d}\boldsymbol{x} &= \int p\|\nabla \log f\|^2 \mathrm{d}\boldsymbol{x} - 2\sum_{i=1}^{d} \int p\partial_i \log r \partial_i \log f \mathrm{d}\boldsymbol{x} + C \\
&= \int p\|\nabla \log f\|^2 \mathrm{d}\boldsymbol{x} + 2\sum_{i=1}^{d} \int q\partial_i r \partial_i \log f \mathrm{d}\boldsymbol{x} + C \\
&= \int p\|\nabla \log f\|^2 \mathrm{d}\boldsymbol{x} + 2\sum_{i=1}^{d} \int r \cdot \partial_i (q\partial_i \log f) \mathrm{d}\boldsymbol{x} + C, \qquad (23) \\
&= \int p\|\nabla \log f\|^2 \mathrm{d}\boldsymbol{x} + 2\sum_{i=1}^{d} \int r \cdot q \cdot \partial_i^2 \log f \mathrm{d}\boldsymbol{x} + 2\sum_{i=1}^{d} \int r \cdot \partial_i q \cdot \partial_i \log f \mathrm{d}\boldsymbol{x} + C \qquad (24)
\end{aligned}$$

and to get (23) we applied integration by parts, where we assumed $q \cdot r \cdot \partial_i \log f \to 0$ as $x_i \to \infty$. In (24), the third term is not tractable due to the lack of information about r and q. Changing the objective to $\int p\|\nabla \log r - \nabla \log f\|^2 \mathrm{d}\boldsymbol{x}$ would also yield an intractable objective for a similar reason.

L. Discussion: Gradient Flow Estimation in Feature Space

One of the issues of local estimation is the *curse of dimensionality*: Local approximation does not work well in high dimensional spaces. However, since the f-divergence gradient flow is always associated with the density ratio function, we can utilize special structures in density ratio functions to estimate $\nabla(h \circ r)(\boldsymbol{x}^\star)$ more effectively.

L.1. Density Ratio Preserving Map

Let $\boldsymbol{s}(\boldsymbol{x})$ be a measurable function, where $\boldsymbol{s} : \mathbb{R}^d \to \mathbb{R}^m, m \leq d$. Consider two random variables, X_p and X_q, each associated with probability density functions p and q, respectively. Define p° and q° as the probability density functions of the random variables $S_p := \boldsymbol{s}(X_p)$ and $S_q := \boldsymbol{s}(X_q)$.

Definition L.1. $\boldsymbol{s}(\boldsymbol{x})$ is a density ratio preserving map if and only if it satisfies the following equality

$$r(\boldsymbol{x}) = r^\circ(\boldsymbol{s}(\boldsymbol{x})), \text{where } r^\circ(\boldsymbol{s}(\boldsymbol{x})) := \frac{p^\circ(\boldsymbol{s}(\boldsymbol{x}))}{q^\circ(\boldsymbol{s}(\boldsymbol{x}))}, \quad \forall \boldsymbol{x} \in \mathcal{X}.$$

We can leverage the density ratio preserving map to reduce the dimensionality of gradient flow estimation. Suppose $\boldsymbol{s}(\boldsymbol{x})$ is a known density ratio preserving map. Define $\boldsymbol{z} := \boldsymbol{s}(\boldsymbol{x})$ and $\boldsymbol{z}^\star := \boldsymbol{s}(\boldsymbol{x}^\star)$. We can see that

$$\underbrace{\nabla(h \circ r)(\boldsymbol{x}^\star)}_{\mathbb{R}^d \mapsto \mathbb{R}^d} = \nabla(h \circ r^\circ)(\boldsymbol{s}(\boldsymbol{x}^\star)) = \underbrace{\nabla^\top \boldsymbol{s}(\boldsymbol{x}^\star)}_{\mathbb{R}^d \mapsto \mathbb{R}^{d \times m}, \text{ known}} \underbrace{\nabla(h \circ r^\circ)(\boldsymbol{z}^\star)}_{\mathbb{R}^d \mapsto \mathbb{R}^m}. \qquad (25)$$

If we can evaluate $\nabla \boldsymbol{s}(\boldsymbol{x}^\star)$, we only need to estimate an m-dimensional gradient $\nabla(h \circ r^\circ)(\boldsymbol{z}^\star)$, which is potentially easier than estimating the original d-dimensional gradient $\nabla(h \circ r)(\boldsymbol{x}^\star)$ using a local linear model.

While Definition L.1 might suggest that $\boldsymbol{s}$ is a very specific function, the requirement for $\boldsymbol{s}$ to preserve the density ratio is quite straightforward. Specifically, $\boldsymbol{s}$ must be *sufficient in expressing the density ratio function.* This requirement is formalized in the following proposition:

Proposition L.2. *Consider a function $\boldsymbol{s} : \mathbb{R}^d \to \mathbb{R}^m$. If there exists a function $g : \mathbb{R}^m \to \mathbb{R}_+$ such that $r(\boldsymbol{x}) = g(\boldsymbol{s}(\boldsymbol{x})), \forall \boldsymbol{x} \in \mathcal{X}$ holds, then $\boldsymbol{s}$ is a density ratio preserving map. Additionally, it follows that $g \circ \boldsymbol{s} = r = r^\circ \circ \boldsymbol{s} \implies g = r^\circ$.*

The proof can be found in Section M.

Proposition L.2 implies that we can identify the density ratio preserving map $\boldsymbol{s}$ by simply learning the ratio function r and using the trained feature transform function as $\boldsymbol{s}$. For instance, in the context of a neural network used to estimate r, $\boldsymbol{s}$ could correspond to the functions represented by the penultimate layer of the network. After identifying $\boldsymbol{s}$, we can simply translate a high dimensional gradient flow estimation into a low dimensional problem according to (25).

Algorithm 1 Searching for a Density Ratio Preserving Map $\boldsymbol{s}$

Inputs: $\mathcal{D}_p$, $\mathcal{D}_q$ and an initial guess of $\hat{\boldsymbol{s}}$.
while $\hat{\boldsymbol{s}}$ not converged **do**
  for each $\boldsymbol{x} \in \mathcal{D}_p \cup \mathcal{D}_q$ **do**
    $\boldsymbol{z} := \hat{\boldsymbol{s}}(\boldsymbol{x})$
    $\left(\hat{\boldsymbol{w}}(\boldsymbol{z}), \hat{b}(\boldsymbol{z})\right) := \underset{\boldsymbol{w}\in\mathbb{R}^m, b\in\mathbb{R}}{\operatorname{argmin}} \ \ell(\boldsymbol{w}, b; \boldsymbol{z}, \mathcal{D}_{p^\circ}, \mathcal{D}_{q^\circ}) \mid_{\boldsymbol{s}=\hat{\boldsymbol{s}}}$
    $\hat{d}(\boldsymbol{z}) := \langle \hat{\boldsymbol{w}}(\boldsymbol{z}), \boldsymbol{z}\rangle + \hat{b}(\boldsymbol{z})$
  end for
  $\hat{\boldsymbol{s}} := \underset{\boldsymbol{s}\in\mathbb{S}}{\operatorname{argmax}} \widehat{\mathbb{E}}_p\left[\hat{d}(\boldsymbol{s}(\boldsymbol{x}))\right] - \widehat{\mathbb{E}}_q\left[\psi_{\mathrm{con}}\left(\hat{d}(\boldsymbol{s}(\boldsymbol{x}))\right)\right]$
end while
Output: $\hat{\boldsymbol{s}}$.

In practice, we find this method works well. However, this approach still requires us to estimate a high dimensional density ratio function r to obtain a feature map $\boldsymbol{s}$.

In the next section, we propose an algorithm of learning $\boldsymbol{s}$ from data without estimating a high dimensional density ratio function.

L.2. Finding Density Ratio Preserving Map

Theorem L.3. *Suppose h is associated with an f-divergence D_ϕ according to Theorem 2.1 and D_ψ is the mirror of D_ϕ. If $r(\boldsymbol{x}) = g^*(\boldsymbol{s}^*(\boldsymbol{x}))$, then $\boldsymbol{s}^*$ must be an* $\arg\sup$ *of the following objective:*

$$\sup_{\boldsymbol{s}} \mathbb{E}_p\left[h(r^\circ(\boldsymbol{s}(\boldsymbol{x})))\right] - \mathbb{E}_q\left[\psi_{\mathrm{con}}(h(r^\circ(\boldsymbol{s}(\boldsymbol{x}))))\right]. \tag{26}$$

Proof. Since $r(\boldsymbol{x}) = g^*(\boldsymbol{s}^*(\boldsymbol{x}))$, Proposition C.1 implies that $(g^*, \boldsymbol{s}^*)$ is necessarily an $\arg\sup$ to the following optimization problem:

$$\sup_{\boldsymbol{s}, g} \mathbb{E}_p\left[h(g(\boldsymbol{s}(\boldsymbol{x})))\right] - \mathbb{E}_q\left[\psi_{\mathrm{con}}(h(g(\boldsymbol{s}(\boldsymbol{x}))))\right].$$

Due to the law of unconscious statistician, $\mathbb{E}_p[f(\boldsymbol{s}(\boldsymbol{x}))] = \mathbb{E}_{p^\circ}[f(\boldsymbol{z})]$, where $\boldsymbol{z} = \boldsymbol{s}(\boldsymbol{x})$. The above optimization problem can be rewritten as

$$\sup_{\boldsymbol{s}} \sup_{g} \mathbb{E}_{p^\circ}\left[h(g(\boldsymbol{z}))\right] - \mathbb{E}_{q^\circ}\left[\psi_{\mathrm{con}}(h(g(\boldsymbol{z})))\right],$$

Proposition C.1 states that for all $\boldsymbol{s}$, $g = r^\circ$ is an $\arg\sup$ of the inner optimization problem. Substituting this optimal solution of g and rewriting the expectation using $\boldsymbol{x}$ again, we arrive

$$\sup_{\boldsymbol{s}} \mathbb{E}_p\left[h(r^\circ(\boldsymbol{s}(\boldsymbol{x})))\right] - \mathbb{E}_q\left[\psi_{\mathrm{con}}(h(r^\circ(\boldsymbol{s}(\boldsymbol{x}))))\right].$$

□

Both expectations in (26) can be approximated using samples from p and q. Given a fixed $\boldsymbol{s}$, $h(r^\circ(\boldsymbol{z}))$ can be approximated by an m-dimensional local linear interpolation

$$h(r^\circ(\boldsymbol{z})) \approx \hat{d}(\boldsymbol{z}) := \langle \hat{\boldsymbol{w}}(\boldsymbol{z}), \boldsymbol{z}\rangle + \hat{b}(\boldsymbol{z}), \left(\hat{\boldsymbol{w}}(\boldsymbol{z}), \hat{b}(\boldsymbol{z})\right) := \underset{\boldsymbol{w}\in\mathbb{R}^m, b\in\mathbb{R}}{\operatorname{argmin}} \ \ell(\boldsymbol{w}, b; \boldsymbol{z}, \mathcal{D}_{p^\circ}, \mathcal{D}_{q^\circ}), \tag{27}$$

where $\mathcal{D}_{p^\circ}, \mathcal{D}_{q^\circ}$ are sets of samples from p° and q° respectively.

Approximating expectations in (26) with samples in $\mathcal{D}_p$ and $\mathcal{D}_q$ and replacing $h \circ r^\circ$ with $\hat{d}$, we solve the following optimization to obtain an estimate of $\boldsymbol{s}$:

$$\hat{\boldsymbol{s}} := \underset{\boldsymbol{s}\in\mathbb{S}}{\operatorname{argmax}} \frac{1}{n_p}\sum_{i=1}^{n_p} \hat{d}\left[\boldsymbol{s}\left(\boldsymbol{x}_p^{(i)}\right)\right] - \frac{1}{n_q}\sum_{i=1}^{n_q} \psi_{\mathrm{con}}\left[\hat{d}\left(\boldsymbol{s}(\boldsymbol{x}_q^{(i)})\right)\right]. \tag{28}$$

The optimization of (28) is a bi-level optimization problem as $\hat{d}$ depends on (27). We propose to divide the whole problem into two steps: First, let $\boldsymbol{s} = \hat{\boldsymbol{s}}$ and solve for $(\hat{\boldsymbol{w}}, \hat{b})$. Then, with the estimated $(\hat{\boldsymbol{w}}, \hat{b})$, we solve for $\hat{\boldsymbol{s}}$. Repeat the above procedure until convergence. This algorithm is detailed in Algorithm 1.

In practice, we restrict $\mathbb{S}$ to be the set of all linear maps via a matrix $\boldsymbol{S} \in \mathbb{R}^{d\times m}$ whose columns are orthonormal basis, i.e., $\boldsymbol{s}(\boldsymbol{x}) := \boldsymbol{S}^\top \boldsymbol{x}$. The Jacobian $\nabla \boldsymbol{s}(\boldsymbol{x})$ is simply $\boldsymbol{S}^\top$.

After obtaining $\hat{\boldsymbol{s}}$, we can approximate the gradient flow using the chain rule described in (25):

$$\nabla(h \circ r)(\boldsymbol{x}^\star) \approx \nabla^\top \hat{\boldsymbol{s}}(\boldsymbol{x}^\star)\hat{\boldsymbol{w}}(\boldsymbol{z}^\star),$$

where $\hat{\boldsymbol{w}}(\boldsymbol{z}^\star)$ is approximated by an m-dimensional local linear interpolation

$$\left(\hat{\boldsymbol{w}}(\boldsymbol{z}^\star), \hat{b}(\boldsymbol{z}^\star)\right) := \underset{\boldsymbol{w}\in\mathbb{R}^m, b\in\mathbb{R}}{\operatorname{argmin}} \ \ell(\boldsymbol{w}, b; \boldsymbol{z}^\star, \mathcal{D}_{p^\circ}, \mathcal{D}_{q^\circ}) \mid_{\boldsymbol{s}=\hat{\boldsymbol{s}}} .$$

M. Discussion: Sufficient Condition of Density Ratio Preserving Map

In this Section, we provide a sufficient condition for $\boldsymbol{s}$ to be a density ratio preserving map.

Lemma M.1. *If there exists some $g : \mathbb{R}^m \to \mathbb{R}_+$, such that $r(\boldsymbol{x}) = g(\boldsymbol{s}(\boldsymbol{x}))$, then $\boldsymbol{s}$ is a density ratio preserving map and $g(\boldsymbol{s}(\boldsymbol{x})) = r^\circ(\boldsymbol{s}(\boldsymbol{x}))$.*

Proof. The statement $g(\boldsymbol{s}(\boldsymbol{x}))$ being a density ratio $\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}$ is equivalent to asserting that $\mathrm{KL}[p, q\cdot(g\circ\boldsymbol{s})] = 0$. Since KL divergence is always non-negative, it means

$$g = \underset{g:\int q(\boldsymbol{x})g(\boldsymbol{s}(\boldsymbol{x}))\mathrm{d}\boldsymbol{x}=1, g\geq 0}{\operatorname{argmin}} \mathbb{E}_p\left[\log\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})g(\boldsymbol{s}(\boldsymbol{x}))}\right] = \mathbb{E}_q[\log g(\boldsymbol{s}(\boldsymbol{x}))] + C_1, \tag{29}$$

i.e., g is a minimizer of $\mathrm{KL}[p, q\cdot(g\circ\boldsymbol{s})]$, where $g \geq 0$ is constrained in a domain where $q\cdot(g\circ\boldsymbol{s})$ is normalized to 1.

Similarly, $r^\circ(\boldsymbol{z})$ being a density ratio $\frac{p^\circ(\boldsymbol{z})}{q^\circ(\boldsymbol{z})}$ is the same as asserting that $\mathrm{KL}[p^\circ, q^\circ r^\circ] = 0$ and is equivalent to

$$r^\circ = \underset{g:\int q^\circ(\boldsymbol{z})g(\boldsymbol{z})\mathrm{d}\boldsymbol{z}=1, g\geq 0}{\operatorname{argmin}} \mathbb{E}_{p^\circ}\left[\log\frac{p^\circ(\boldsymbol{z})}{q^\circ(\boldsymbol{z})g(\boldsymbol{z})}\right] = \mathbb{E}_{q^\circ}[\log g(\boldsymbol{z})] + C_2, \tag{30}$$

i.e., r° is a minimizer of $\mathrm{KL}[p^\circ, q^\circ g]$.

In fact, one can see that (29) and (30) are identical optimization problems due to the law of the unconscious statistician: $\int q(\boldsymbol{x})g(\boldsymbol{s}(\boldsymbol{x}))\mathrm{d}\boldsymbol{x} = \mathbb{E}_q[g(\boldsymbol{s}(\boldsymbol{x}))] = \mathbb{E}_{q^\circ}[(g(\boldsymbol{z}))] = \int q^\circ(\boldsymbol{z})g(\boldsymbol{z})\mathrm{d}\boldsymbol{z}$ and $\mathbb{E}_q[\log g(\boldsymbol{s}(\boldsymbol{x}))] = \mathbb{E}_{q^\circ}[\log g(\boldsymbol{z})]$, which means their solution sets are the same. Therefore, for any g that minimizes (29), it must also minimize (30). Hence it satisfies the following equality $r^\circ(\boldsymbol{s}(\boldsymbol{x})) = g(\boldsymbol{s}(\boldsymbol{x})) = r(\boldsymbol{x})$, where the second equality is by our assumption. □

N. Discussion: Stein Variational Gradient Descent

SVGD minimizes $\mathrm{KL}[q_{t+1}, p]$, where samples of q_{t+1} is constructed using the following deterministic rule:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \eta\boldsymbol{u}_t(\boldsymbol{x}_t). \tag{31}$$

$\boldsymbol{x}_t$ are particles at iteration t, $\boldsymbol{u}_t \in \mathcal{H}^d$, a d-dimensional Reproducing Kernel Hilbert Space (RKHS) with a kernel function $l(\boldsymbol{x}, \boldsymbol{x}')$. Liu and Wang (2016) shows the optimal update $\boldsymbol{u}_t$ has a closed form:

$$\boldsymbol{u}_t^{\mathrm{svgd}} := \mathbb{E}_{q_t}[l(\boldsymbol{x}_t, \cdot)\nabla\log p(\boldsymbol{x}_t) + \nabla l(\boldsymbol{x}_t, \cdot)]. \tag{32}$$

In practice, expectations $\mathbb{E}_{q_t}[\cdot]$ can be approximated by $\widehat{\mathbb{E}}_{q_t}[\cdot]$, i.e., the sample average taken from the particles at time t.

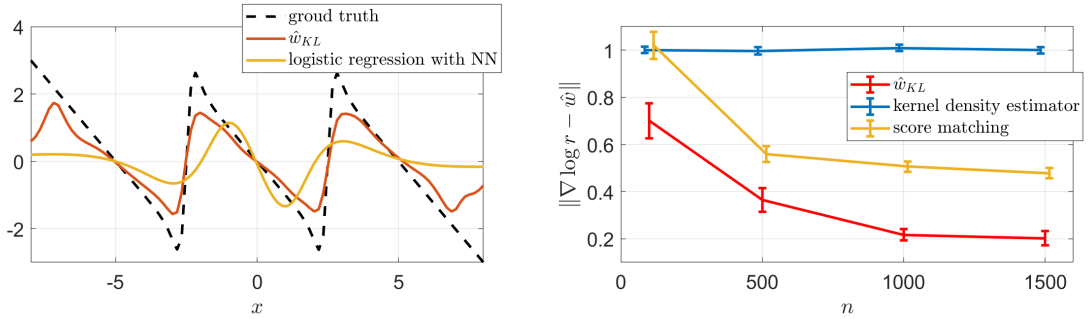


Figure 9. Left: $\nabla \log r(\boldsymbol{x})$ and its approximations. Right: Estimation error with standard error.

Chewi et al. (2020) links SVGD with f-divergence WGF: $\boldsymbol{u}_t^{\text{svgd}}$ is *the backward KL divergence WGF* under the coordinate-wise transform of an integral operator. Indeed, the i-th dimension of SVGD update $u_{t,i}^{\text{svgd}}$ can be expressed as

$$\begin{aligned} u_{t,i}^{\text{svgd}} :=& \mathbb{E}_{q_t}\left[l(\boldsymbol{x}_t,\cdot)\partial_i \log p(\boldsymbol{x}_t)\right] + \mathbb{E}_{q_t}\left[\partial_i l(\boldsymbol{x}_t,\cdot)\right] \\ =& \mathbb{E}_{q_t}\left[l(\boldsymbol{x}_t,\cdot)\partial_i \log p(\boldsymbol{x}_t)\right] - \mathbb{E}_{q_t}\left[l(\boldsymbol{x}_t,\cdot)\partial_i \log q_t(\boldsymbol{x}_t)\right] \\ =& \int q_t(\boldsymbol{x}_t) l(\boldsymbol{x}_t,\cdot)\partial_i \log \frac{p(\boldsymbol{x}_t)}{q_t(\boldsymbol{x}_t)} \mathrm{d}\boldsymbol{x}_t, \end{aligned} \tag{33}$$

where the last line is an integral operator (Wainwright, 2019) of the functional $\partial_i \log \frac{p}{q_t}$, i.e., the i-th dimension of the backward KL divergence flow $\nabla \log r_t$. [4] Due to the reproducing property of RKHS, the SVGD update at some fixed point $\boldsymbol{x}^\star$ can be written as

$$\boldsymbol{u}^{\text{svgd}}(\boldsymbol{x}^\star) = \langle \boldsymbol{u}_t^{\text{svgd}}, l(\cdot,\boldsymbol{x}^\star)\rangle_{\mathcal{H}^d} = \mathbb{E}_q[l(\boldsymbol{x},\boldsymbol{x}^\star)\nabla \log r(\boldsymbol{x})].$$

O. Additional Experiments

O.1. Gradient Estimation

Now we investigate the performance of estimating $\nabla \log r(\boldsymbol{x})$ using the proposed gradient estimator and an indirect estimator using logistic regression. For the indirect estimator, we first train a Multilayer Perceptron (MLP) using a binary logistic regression to approximate $\log r$. Then obtain $\nabla \log r$ by auto-differentiating the estimated log ratio. The kernel bandwidth in our method is tuned by using the model selection criterion described in Section 4.4.

To conduct the experiments, we let $p = \mathcal{N}(-5,.5)/3 + \mathcal{N}(0,.5)/3 + \mathcal{N}(5,.5)/3$ and $q = \mathcal{N}(-5,1)/3 + \mathcal{N}(0,1)/3 + \mathcal{N}(5,1)/3$. From each distribution, 5000 samples are generated for approximating the gradients.

The left plot in Figure 9 shows the true gradient and its approximations. It can be seen that the direct gradient estimation is more accurate than estimating the log ratio first then taking the gradient.

The right plot in Figure 9 displays the estimation errors of different methods, comparing the proposed method with Kernel Density Estimation (KDE) and score matching, all applied to the same distributions in the previous experiment. KDE was previously used in approximating WGF (Wang et al., 2022). It first estimates p and q with $\hat{p}$ and $\hat{q}$ separately using non-parametric kernel density estimators, then approximates $\nabla \log r$ with $\nabla \log \hat{p} - \nabla \log \hat{q}$. The score matching approximates $\nabla \log \hat{p}$ and $\nabla \log \hat{q}$ with the minimizers of Fisher-divergence. It has also been used in simulating particle ODEs in a previous work (Maoutsa et al., 2020). The estimation error plot shows that the proposed estimator yields more accurate results compared to the other two kernel-based gradient estimation methods, namely KDE and score matching.

O.2. Wasserstein Gradient Flow

In this experiment, we test the performance of the proposed gradient estimators by generating samples from a high dimensional target distribution (MNIST handwritten digits). We will check whether the quality of the particles can be improved by performing WGC using our estimated updates. Note that we do not intend to compare the generated samples

[4] Note that the second equality in (33) is due to the integration by parts, and only holds under conditions that $\lim_{\|\boldsymbol{x}\|\to\infty} q_t(\boldsymbol{x}) = 0$.

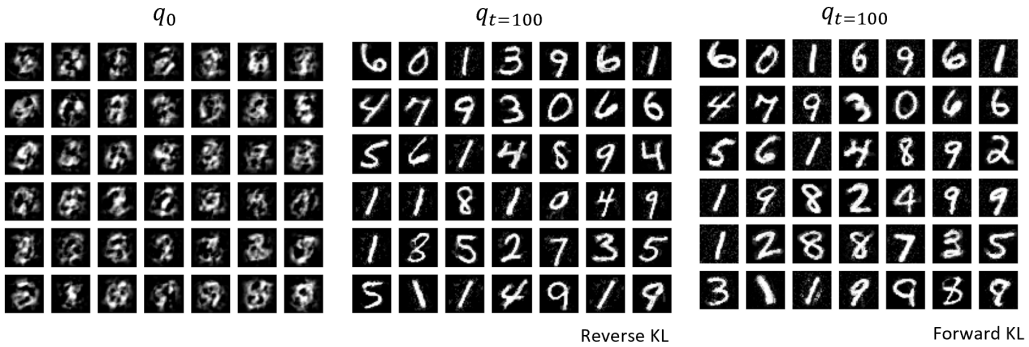


Figure 10. Samples generated using forward and backward KL velocity field.

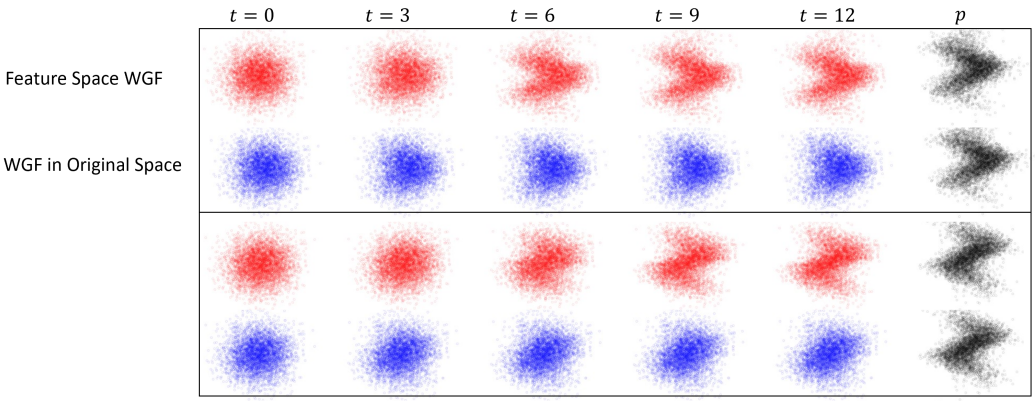


Figure 11. $\{\boldsymbol{x}_{q_t}\}$ in the first two dimensions x_1, x_2. Feature space WGF converges in fewer steps compared to WGF in the original space. Above: $g = \cos$. Below: $g = \sin$.

with NN-based approaches, as the focus of our paper is on local estimation using kernel functions. We perform two different WGFs, forward KL and backward KL whose updates are approximated using $\hat{\boldsymbol{w}}_{\rightarrow}$ and $\hat{\boldsymbol{w}}_{\leftarrow}$ respectively. We let the initial particle distribution q_0 be $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ are the mean and covariance of the target dataset and fix the kernel bandwidth using "the median trick" (see the appendix for details).

The generated samples together with samples from the initial distribution q_0 are shown in Figure 10. Judging from the generated sample quality, it can be seen that both VGDs perform well and both have made significant improvements from the initial samples drawn from q_0.

We also provide two videos showing the animation of the first 100 gradient steps:

- Forward KL: `https://youtube.com/shorts/HZcvUykrpbc`
- Backward KL: `https://youtube.com/shorts/AgN6dsDecCM`

O.3. Feature Space Wasserstein Gradient Flow

In this experiment, we run WGF in a 5-dimensional space. The target distribution is

$$p = p_{12}(\boldsymbol{x}_{1,2}) p_{345}(\boldsymbol{x}_{3,4,5}), p_{345} := \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

and p_{12} is constructed by a generative process. We generate samples in the first two dimensions as follows

$$X_1 = g(X_2) + \epsilon, X_2 \sim \mathcal{N}(0, 1), \epsilon \sim \mathcal{N}(0, 1).$$

In our experiment, we draw 5000 samples from p, and 5000 samples from $q_0 := \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and run the backward KL gradient flow. Clearly, this WGF has a low dimensional structure since p and q only differs in the first two dimensions. We also run

feature space backward KL field whose updates are calculated using (25). The feature function $\boldsymbol{s}(\boldsymbol{x}) = \boldsymbol{S}^\top \boldsymbol{x}$ is learned by Algorithm 1.

The resulting particle evolution for both processes is plotted in Figure 11. For visualization purposes, we only plot the first two dimensions. It can be seen that the particles converge much faster when we explicitly exploit the subspace structure using the feature space WGF. In comparison, running WGF in the original space converges at a much slower speed.

In this experiments, we set the learning rates for both WGF and feature space WGF to be 0.1 and the kernel bandwidth σ in our local estimators is tuned using cross validation with a candidate set ranging from 0.1 to 2.