GRADIENT FLOWS ON THE FEATURE-GAUSSIAN MANIFOLD

Anonymous authors
Paper under double-blind review

ABSTRACT

The scarcity of labeled data is a long-standing challenge for cross-domain machine learning tasks. This paper leverages the existing dataset (i.e., source) to augment new samples that are close to the dataset of interest (i.e., target). To relieve the need to learn a metric on the feature-label space, we lift both datasets to the space of probability distributions on the feature-Gaussian manifold, and then develop a gradient flow that minimizes the maximum mean discrepancy loss. To perform the gradient flow of distributions on the curved feature-Gaussian space, we unravel the Riemannian structure of the space and compute explicitly the Riemannian gradient of the loss function induced by the optimal transport metric. For practical purposes, we also propose a discretized flow, and provide conditional results guaranteeing the global convergence of the flow to the optimum. We illustrate the results of our proposed gradient flow method in several real-world datasets.

1 INTRODUCTION

A major challenge facing machine learning and data science is the lack of labeled data. A popular approach is developing learning methods that can interpolate, adapt, or transfer knowledge across datasets and domains. Some well-known methods for these tasks are domain adaptation (Ben-David et al., 2007; Mansour et al., 2009; Courty et al., 2017; Damodaran et al., 2018; Gong et al., 2012; Taigman et al., 2016), transfer learning (Long et al., 2017; Pan & Yang, 2010; Zamir et al., 2018), and meta-learning (Finn et al., 2017; Khodak et al., 2019). Recently, these methods have produced promising results for important tasks in autonomous driving and robotics (Wang et al., 2018; Bousmalis et al., 2018).

A straightforward remedy to the lack of data is to devise mechanisms to synthesize new sensible data samples, particularly in the target domain. In this paper, we consider a specific setup in which we have access to labelled data samples generated from both the source and the target domain. More concretely, we consider a covariate space $\mathcal{X} = \mathbb{R}^m$ and a *categorical* label space $\mathcal{Y}$. We are given a source domain dataset consisting of N samples $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ for $i = 1, \ldots, N$, and a target domain dataset of M samples $(\bar{x}_j, \bar{y}_j) \in \mathcal{X} \times \mathcal{Y}$ for $j = 1, \ldots, M$. We consider the situation where M is not large enough, and the target data is scarce. The ultimate goal of this paper is to generate new samples in the target domain, and we aim to create new samples whose distribution is as close as possible to the unknown distribution that governs the target domain.

We here adopt a gradient flow method to synthesize new, unseen data samples. Because we have access to the source domain samples, it is possible to flow each source sample towards the target data in order to minimize a certain loss function. If the loss function is chosen to reflect the dissimilarity between distributions, and if the flow is properly designed to converge, then the terminal product of the flow will provide us with new samples that can sufficiently approximate the data-generating distribution of the target domain. As a consequence, using gradient flows is a sensible approach to synthesize target domain samples.

Unfortunately, formulating a gradient flow algorithm for labelled data with categorical set $\mathcal{Y}$ is problematic. Indeed, there is no clear metric structure on $\mathcal{Y}$ in order to define the topological neighborhood, this in turn leads to the difficulty of forming the gradients with respect to the categorical component. To overcome this difficulty, a gradient flow on the dataset space was recently proposed in Alvarez-Melis & Fusi (2021) by leveraging a new notion of distance between datasets in Alvarez-Melis & Fusi (2020); Courty et al. (2017); Damodaran et al. (2018). The main idea behind this approach is to reparametrize the categorical space $\mathcal{Y}$ using the conditional distribution of the features, which is assumed to be Gaussian, and then construct a gradient flow on the feature-Gaussian space. Nevertheless, the theoretical analysis in Alvarez-Melis & Fusi (2021) focuses solely on the gradients with respect to the feature, and there is no derivation of the flow with respect to the Gaussian component. In fact, the space of Gaussian distributions is not a (flat) vector space, and extracting gradient information depends on the choice of the metric imposed on this Gaussian space.

Contributions. We study in this paper a gradient flow approach to synthesize new labelled samples related to the target domain. To construct this flow, we consider the space of probability distributions on the feature-Gaussian manifold, and we are metrizing this space with an optimal transport distance. We summarize the contributions of this paper as follows.

- We study in details the Riemannian structure of the feature-Gaussian manifold in Section 3.1, as well as the Riemannian structure of the space of probability measures supported on this manifold in Section 3.2.
- We consider a gradient flow that minimizes the squared maximum mean discrepancy (MMD) loss function to the target distribution. We describe explicitly the (Riemannian) gradient of the squared MMD in Lemma 4.1, and we provide a partial differential equation describing the evolution of the gradient flow that follows the (Riemannian) steepest descent direction.
- We propose two discretization schemes to approximate the continuous gradient flow equation: an Euler scheme in Section 4.1 and an Euler scheme with noise in Section 4.2. We provide conditions guaranteeing the global convergence of our gradient flows to the optimum in both continuous and discretized schemes.

Our gradient flows minimize the MMD loss function, thus it belongs to the family of MMD gradient flows that was pioneered in Mroueh et al. (2019) and Arbel et al. (2019), and further extended in Mroueh & Nguyen (2021). The MMD function compares two distributions via their kernel mean embeddings on a *flat* reproducing kernel Hilbert space (RKHS). In contrast to the Kullback-Leibler divergence flow, the MMD flow can employ a sample approximation for the target distribution (Liu, 2017). Further, the squared MMD possesses unbiased sample gradients (Bińkowski et al., 2018; Bellemare et al., 2017). However, existing literature on MMD flows focus on distributions on (flat) Euclidean spaces. The flow developed in our paper here is for distributions on the (curved) Riemannian feature-Gaussian space. Moreover, our approach is distinctive from the flow in Alvarez-Melis & Fusi (2021) because the flow therein does not consider the gradient in the Gaussian component. Here, we impose a specific metric on the Gaussian component, and we compute explicitly the (Riemannian) gradient of the MMD loss function with respect to this metric to formulate our flow.

Generating new data samples is particularly useful when we have to train classifiers with limited labelled target data. The numerical experiments in Section 5 will demonstrate that our gradient flows on the feature-Gaussian manifold can effectively augment the target data in the few-shot learning setting, and thus can significantly boost the accuracy in the classification task.

Other related works. Nonparametric gradient flows using the 2-Wasserstein distance between distributions are investigated in (Ambrosio et al., 2008; Jordan et al., 1998; Otto, 2001; Villani, 2008; Santambrogio, 2015; 2017; Frogner & Poggio, 2020; Kolouri et al., 2020), but only for distributions on Euclidean spaces and for different loss functions. Related nonparametric gradient flows with other metrics include Sliced-Wasserstein Descent (Liutkus et al., 2019), Stein Descent (Liu, 2017; Liu & Wang, 2016; Duncan et al., 2019), and Sobolev Descent (Mroueh et al., 2019), however, they also consider only distributions on Euclidean spaces. In particular, (Liu, 2017; Duncan et al., 2019) introduce Riemannian structures for the Stein geometry on flat spaces, while ours is for an optimal transport metric on a curved space. On the other hand, related parametric flows for training generative adversarial networks have been studied in (Bottou et al., 2017; Chizat & Bach, 2018; Chen & Li, 2018; Arbel et al., 2020; Chizat, 2020; Mroueh & Nguyen, 2021).

Notations. We use $\mathbb{S}^n$ to denote the set of $n \times n$ real and symmetric matrices, and $\mathbb{S}^n_{++} \subset \mathbb{S}^n$ consists of all positive definite matrices. For $A \in \mathbb{S}^n$, $\mathrm{tr}(A) := \sum_i A_{ii}$. We use $\langle \cdot, \cdot \rangle$ and $\| \cdot \|_2$ to denote the standard inner product and norm on Euclidean spaces. Let $\mathcal{P}(X)$ be the collection of all probability distributions with finite second moment on metric space X. If $\varphi : X \to Y$ is a Borel map and $\nu \in \mathcal{P}(X)$, then the push-forward $\varphi_\# \nu$ is the distribution on Y given by $\varphi_\# \nu(E) = \nu(\varphi^{-1}(E))$ for all Borel sets $E \subset Y$. For a function f of the continuous time variable t, f_t denotes the value of f at t while $\partial_t f$ denotes the standard derivative of f w.r.t. t. Also, δ_z denotes the Dirac delta measure at z.

2 Labelled Data Synthesis via Gradient Flows of Lifted Distributions

In this section, we describe our approach to synthesize target domain samples using gradient flows. A holistic view of our method is presented in Figure 1.

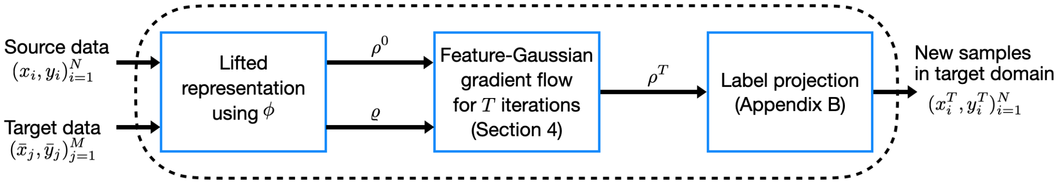
Source data
$(x_i, y_i)_{i=1}^N$
Target data
$(\bar{x}_j, \bar{y}_j)_{j=1}^M$
Lifted representation using ϕ
ρ^0
ϱ
Feature-Gaussian gradient flow for T iterations (Section 4)
ρ^T
Label projection (Appendix B)
New samples in target domain
$(x_i^T, y_i^T)_{i=1}^N$

Figure 1: Schematic view of our approach: The source and target datasets are first lifted to distributions ρ^0 and ϱ on the feature-Gaussian space (left box). We then run a gradient flow for T iterations to get a terminal distribution ρ^T (middle). Atoms of ρ^T are projected to get labeled target samples (right).

In the first step, we would need to lift the feature-label space $\mathcal{X} \times \mathcal{Y}$ to a higher dimensional space where a metric can be defined. Consider momentarily the source data samples $(x_i, y_i)_{i=1}^N$. Notice that this data can be represented as an empirical distribution ν on $\mathcal{X} \times \mathcal{Y}$. More precisely, we have $\nu = N^{-1} \sum_{i=1}^N \delta_{(x_i, y_i)}$. Because $\mathcal{Y}$ is discrete, the law of conditional probabilities allows us to identify the conditional distribution ν_y of $X|Y = y$ under ν. The lifting procedure is obtained by employing a pre-determined mapping $\phi : \mathcal{X} \to \mathbb{R}^n$, and any categorical value $y \in \mathcal{Y}$ can now be represented as an n-dimensional distribution $\phi_\# \nu_y$. Using this lifting, any source sample $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ is lifted to a point $(x_i, \phi_\# \nu_{y_i}) \in \mathcal{X} \times \mathcal{P}(\mathbb{R}^n)$ and the source dataset is representable as an empirical distribution of the form $N^{-1} \sum_{i=1}^N \delta_{(x_i, \phi_\# \nu_{y_i})}$.

The lifted representation of a categorical value $y \in \mathcal{Y}$ as an n-dimensional distribution $\phi_\# \nu_y \in \mathcal{P}(\mathbb{R}^n)$ is advantageous because $\mathcal{P}(\mathbb{R}^n)$ is metrizable, for example, using the 2-Wasserstein distance. The downside, unfortunately, is that $\mathcal{P}(\mathbb{R}^n)$ is infinite dimensional, and encoding the datasets in this lifted representation is not efficient. To resolve this issue, we assume that $\phi_\# \nu_y$ is Gaussian for all $y \in \mathcal{Y}$, and thus any distribution $\phi_\# \nu_y$ can be characterized by the mean vector $\mu_y \in \mathbb{R}^n$ and covariance matrix $\Sigma_y \in \mathbb{S}_{++}^n$ defined as

$$\mu_y = \int_{\mathcal{X}} \phi(x) \nu_y(\mathrm{d}x), \qquad \Sigma_y = \int_{\mathcal{X}} \phi(x)\phi(x)^\top \nu_y(\mathrm{d}x) - \mu_y \mu_y^\top \qquad \forall y \in \mathcal{Y}.$$

In real-world settings, the conditional moments of $\phi(X)|Y$ are sufficiently different for $y \neq y'$, and thus the representations using (μ_y, Σ_y) will unlikely lead to any loss of label information. With this lifting, the source data thus can be represented as an empirical distribution ρ^0 on $\mathbb{R}^m \times \mathbb{R}^n \times \mathbb{S}_{++}^n$ via

$$\rho^0 = N^{-1} \sum\nolimits_{i=1}^N \delta_{(x_i, \mu_{y_i}, \Sigma_{y_i})}.$$

By an analogous construction to compute $\bar{\mu}_y$ and $\bar{\Sigma}_y$ using the target data, the target domain data $(\bar{x}_j, \bar{y}_j)_{j=1}^M$ can be represented as another empirical distribution

$$\varrho = M^{-1} \sum\nolimits_{j=1}^M \delta_{(\bar{x}_j, \bar{\mu}_{\bar{y}_j}, \bar{\Sigma}_{\bar{y}_j})}.$$

Let us denote the shorthand $\mathcal{Z} = \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{S}_{++}^n$, then ρ^0 and ϱ are both probability measures on $\mathcal{Z}$. We refer to ρ^0 and ϱ as the feature-Gaussian representations of the source and target datasets.

We now consider the gradient flow associated with the optimization problem

$$\min_{\rho \in \mathcal{P}(\mathcal{Z})} \left\{ \mathcal{F}(\rho) := \frac{1}{2} \mathrm{MMD}(\rho, \varrho)^2 \right\}$$

under the initialization $\rho = \rho^0$. The objective function $\mathcal{F}(\rho)$ quantifies how far an incumbent solution ρ is from the target distribution ϱ, measured using the MMD distance. In Sections 3 and 4, we will provide the necessary ingredients to construct this flow.

Suppose that after T iterations of the (discretized) gradient flow algorithm, we obtain a distribution $\rho^T \in \mathcal{P}(\mathcal{Z})$ that is sufficiently close to ϱ, i.e., $\mathcal{F}(\rho^T)$ is close to zero. Then we can recover new target samples by projecting the atoms of the distribution ρ^T to the locations on $\mathcal{X} \times \mathcal{Y}$. This projection can be achieved efficiently by solving a linear optimization problem, as suggested in Appendix B.

Remark 2.1 (Reduction of dimensions)**.** *If $m = n$ and ϕ is the identity map, then our lifting procedure coincides with that proposed in Alvarez-Melis & Fusi (2020). However, a large dimension n is redundant, especially when the cardinality of $\mathcal{Y}$ is low. If $n \ll m$, then ϕ offers significant reduction in the number of dimensions, and will speed up the gradient flow algorithms. In this paper, we use ϕ as the t-SNE embedding. According to van der Maaten & Hinton (2008), t-SNE's low-dimensional embedded space forms a Student-t distribution, and SNE uses a Gaussian distribution. Our proposed framework can be straightforwardly extended to elliptical distributions since the Bures distance still has the same closed-form as for the Gaussian distributions (Gelbrich, 1990).*

3 Riemannian Geometry of the Spaces $\mathcal{Z}$ and $\mathcal{P}(\mathcal{Z})$

If we opt to measure the distance between two Gaussian distributions using the 2-Wasserstein metric, then this choice would induce a natural distance d on the space $\mathcal{Z} = \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{S}^n_{++}$ prescribed as

$$d\big((x_1, \mu_1, \Sigma_1), (x_2, \mu_2, \Sigma_2)\big) \coloneqq \big[\|x_1 - x_2\|_2^2 + \|\mu_1 - \mu_2\|_2^2 + \mathbb{B}(\Sigma_1, \Sigma_2)^2\big]^{\frac{1}{2}} \tag{3.1}$$

where $\mathbb{B}$ is the Bures metric on $\mathbb{S}^n_{++}$ given by $\mathbb{B}(\Sigma_1, \Sigma_2) \coloneqq \big[\mathrm{tr}(\Sigma_1 + \Sigma_2 - 2[\Sigma_1^{\frac{1}{2}} \Sigma_2 \Sigma_1^{\frac{1}{2}}]^{\frac{1}{2}})\big]^{\frac{1}{2}}$.

As $\mathbb{B}$ is a metric on $\mathbb{S}^n_+$ (Bhatia et al., 2019, p.167), d is hence a product metric on $\mathcal{Z}$. This section serves two purposes: first, we study the non-Euclidean geometry of $\mathcal{Z}$ under the ground metric d. Second, we investigate the Riemannian structure on $\mathcal{P}(\mathcal{Z})$, the space of all distributions supported on $\mathcal{Z}$ and with finite second moment, that is induced by the optimal transport distance. These Riemannian structures are required to define the Riemannian gradients of any loss functionals on $\mathcal{P}(\mathcal{Z})$, and will play an important role in our development of the gradient flow for the squared MMD.

3.1 Geometry of $\mathcal{Z}$

The space $\mathcal{Z}$ is not a linear vector space. In this section, we reveal the Riemannian structure on $\mathcal{Z}$ associated to the ground metric d. As we shall see, $\mathcal{Z}$ is a curved space as its geodesics are not straight lines and involve solutions to the Lyapunov equation. For any positive definite matrix $\Sigma \in \mathbb{S}^n_{++}$ and any symmetric matrix $V \in \mathbb{S}^n$, the Lyapunov equation

$$H\Sigma + \Sigma H = V \tag{3.2}$$

has a unique solution $H \in \mathbb{S}^n$ (Bhatia, 1997, Theorem VII.2.1). Let $\mathrm{L}_\Sigma[V]$ denote this unique solution H.

Riemannian metric. The space $\mathbb{S}^n_{++}$ is a Riemannian manifold with the Bures metric $\mathbb{B}$ as the associated distance function, see Takatsu (2011, Proposition A). Since $\mathcal{Z}$ is the product of two Euclidean spaces and $\mathbb{S}^n_{++}$, this gives rise to the following geometric structure for $\mathcal{Z}$.

Proposition 3.1 (Geometry of $\mathcal{Z}$)**.** *The space $\mathcal{Z}$ is a Riemannian manifold: at each point $z = (x, \mu, \Sigma) \in \mathcal{Z}$, the tangent space is $\mathrm{T}_z\mathcal{Z} = \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{S}^n$ and the Riemannian metric is given by*

$$\big\langle (w_1, v_1, V_1), (w_2, v_2, V_2) \big\rangle_z \coloneqq \langle w_1, w_2 \rangle + \langle v_1, v_2 \rangle + \langle V_1, V_2 \rangle_\Sigma \tag{3.3}$$

for two tangent vectors (w_1, v_1, V_1) and (w_2, v_2, V_2) in $\mathbb{R}^m \times \mathbb{R}^n \times \mathbb{S}^n$, where $\langle V_1, V_2 \rangle_\Sigma \coloneqq tr\Big(\mathrm{L}_\Sigma[V_1]\, \Sigma\, \mathrm{L}_\Sigma[V_2]\Big)$. Moreover, the distance function corresponding to this Riemannian metric coincides with the distance d given by (3.1).

The proofs of Proposition 3.1 and all other results will be provided in the Appendix A.

Geodesic and exponential map. As $\mathcal{Z}$ is a product Riemannian manifold, any geodesic in $\mathcal{Z}$ is of the form (θ, γ, Γ) with θ, γ being the Euclidean geodesics (straight lines) and Γ being a geodesic in the Riemannian manifold $\mathbb{S}^n_{++}$. More precisely, for each $\Sigma \in \mathbb{S}^n_{++}$ and each tangent vector $V \in \mathbb{S}^n$, the geodesic in the manifold $\mathbb{S}^n_{++}$ emanating from Σ with direction V is given by

$$\Gamma(t) = (I + t\mathrm{L}_\Sigma[V])\Sigma(I + t\mathrm{L}_\Sigma[V]) \quad \text{for } t \in J^*, \tag{3.4}$$

where J^* is the open interval about the origin given by $J^* = \{t \in \mathbb{R} : I + t\mathrm{L}_\Sigma[V] \in \mathbb{S}^n_{++}\}$ (Malagò et al., 2018). As a consequence, for each point $(x, \mu, \Sigma) \in \mathcal{Z}$ and each tangent vector $(w, v, V) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{S}^n$, the Riemannian exponential map in $\mathcal{Z}$ is given by

$$\exp_{(x,\mu,\Sigma)}(t(w, v, V)) := (\theta(t), \gamma(t), \Gamma(t)) \quad \text{for } t \in J^*, \tag{3.5}$$

where $\theta(t) := x + tw$, $\gamma(t) := \mu + tv$, and $\Gamma(t)$ is defined by (3.4). Note that by its definition, $t \mapsto \exp_{(x,\mu,\Sigma)}(t(w, v, V))$ is the geodesic in $\mathcal{Z}$ emanating from (x, μ, Σ) with direction (w, v, V).

Gradient and divergence. Given the Riemannian metric (3.3), ones can define the corresponding notion of gradient and divergence (Lee, 2003). For a differentiable function $\varphi : \mathcal{Z} \to \mathbb{R}$, its gradient $\nabla_d\varphi(z)$ is the unique element in the tangent space $\mathbb{R}^m \times \mathbb{R}^n \times \mathbb{S}^n$ satisfying

$$\big\langle \nabla_d\varphi(z), (w, v, V)\big\rangle_z = D\varphi_z(w, v, V) \quad \text{for all } (w, v, V) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{S}^n$$

with $D\varphi_z(w, v, V)$ denoting the standard directional derivative of φ at z in the direction (w, v, V). By exploiting the special form of $\langle\cdot,\cdot\rangle_z$ in (3.3), we can compute $\nabla_d\varphi(z)$ explicitly:

Lemma 3.2 (Gradients)**.** *For a differentiable function $\varphi : \mathcal{Z} \to \mathbb{R}$, we have for $z = (x, \mu, \Sigma)$ that*

$$\nabla_d\varphi(z) = \Big(\nabla_x\varphi(z),\ \nabla_\mu\varphi(z),\ 2[\nabla_\Sigma\varphi(z)]\Sigma + 2\Sigma[\nabla_\Sigma\varphi(z)]\Big), \tag{3.6}$$

where $(\nabla_x, \nabla_\mu, \nabla_\Sigma)$ are the standard (Euclidean) gradients of the respective components.

The last component in formula (3.6) for $\nabla_d\varphi$ reflects the curved geometry of $\mathcal{Z}$, and can be interpreted as the Riemannian gradient of the function $\Sigma \mapsto \varphi(x, \mu, \Sigma)$ w.r.t. the Bures distance $\mathbb{B}$.

For a continuous vector field $\Phi : \mathcal{Z} \to \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{S}^n$ and a distribution $\rho \in \mathcal{P}(\mathcal{Z})$, the divergence $\mathrm{div}_d(\rho\Phi)$ is defined as the signed measure on $\mathcal{Z}$ satisfying the following integration by parts formula

$$\int_{\mathcal{Z}} \varphi(z)\,\mathrm{div}_d(\rho\Phi)(\mathrm{d}z) = -\int_{\mathcal{Z}} \langle\Phi(z), \nabla_d\varphi(z)\rangle_z\ \rho(\mathrm{d}z)$$

for every differentiable function $\varphi : \mathcal{Z} \to \mathbb{R}$ with compact support. In case ρ has a density w.r.t. the Riemannian volume form on $\mathcal{Z}$, then this definition coincides with the standard divergence operator induced by Riemannian metric (3.3).

3.2 Optimal Transport and Riemannian Structure on $\mathcal{P}(\mathcal{Z})$

To define a gradient low for probability distributions on $\mathcal{Z}$, it is essential to have a concept of gradients for functionals defined on $\mathcal{P}(\mathcal{Z})$. This requires a meaningful Riemannian structure on $\mathcal{P}(\mathcal{Z})$, and here, we adopt a Riemannian structure generated by the optimal transport on $\mathcal{P}(\mathcal{Z})$ with ground cost d^2. The optimal transport metric $\mathbb{W}(\rho_0, \rho_1)$ between any two distributions ρ_0, $\rho_1 \in \mathcal{P}(\mathcal{Z})$ is defined by formula (A.4) of Appendix A.1. As $(\mathcal{Z}, d)$ is a Riemannian manifold by Proposition 3.1, it follows from the celebrated Benamou-Brenier formula (Benamou & Brenier, 2000) that $\mathbb{W}$ can be expressed in terms of a dynamic formulation. Precisely,

$$\mathbb{W}(\rho_0, \rho_1)^2 = \inf_{(\rho,\phi)\in\mathcal{A}(\rho_0,\,\rho_1)} \int_0^1 \int_{\mathcal{Z}} \|\nabla_d\phi_t(z)\|_z^2\ \rho_t(\mathrm{d}z)\,\mathrm{d}t, \tag{3.7}$$

where $\mathcal{A}(\rho_0,\ \rho_1)$ is the collection of all pairs (ρ, ϕ) of curve $\rho : [0, 1] \to \mathcal{P}(\mathcal{Z})$ with endpoints ρ_0 and ρ_1, and function $\phi : [0, 1] \times \mathcal{Z} \to \mathbb{R}$ that satisfies the continuity equation

$$\partial_t\rho + \mathrm{div}_d(\rho_t\nabla_d\phi_t) = 0 \quad \text{in the sense of distributions in} \quad (0, 1) \times \mathcal{Z}. \tag{3.8}$$

Riemannian metric on $\mathcal{P}(\mathcal{Z})$. The formulation (3.7) gives rise to the following Riemannian structure on $\mathcal{P}(\mathcal{Z})$ induced by distance $\mathbb{W}$. First, the continuity equation enables us to identify a tangent vector $\partial_t\rho$ with the divergence $-\mathrm{div}_d(\rho_t\nabla_d\phi_t)$. Thus the tangent space of $\mathcal{P}(\mathcal{Z})$ at a distribution ρ can be defined as $\mathrm{T}_\rho\mathcal{P}(\mathcal{Z}) := \big\{-\mathrm{div}_d(\rho\nabla_d\varphi) :\ \varphi \text{ is a differentiable function with compact support on } \mathcal{Z}\big\}$. Second, we let $g_\rho : \mathrm{T}_\rho\mathcal{P}(\mathcal{Z}) \times \mathrm{T}_\rho\mathcal{P}(\mathcal{Z}) \longrightarrow \mathbb{R}$ be the Riemannian metric tensor given by

$$g_\rho(\zeta_1, \zeta_2) := \int_{\mathcal{Z}} \langle\nabla_d\varphi_1(z), \nabla_d\varphi_2(z)\rangle_z\ \rho(\mathrm{d}z) \tag{3.9}$$

for $\zeta_1 = -\mathrm{div}_d(\rho\nabla_d\varphi_1)$ and $\zeta_2 = -\mathrm{div}_d(\rho\nabla_d\varphi_2)$. With this definition and due to (3.8), formula (3.7) can be rewritten using the metric tensor as $\mathbb{W}(\rho_0, \rho_1)^2 = \inf_{(\rho,\phi)\in\mathcal{A}(\rho_0,\,\rho_1)} \int_0^1 g_{\rho_t}(\partial_t\rho, \partial_t\rho)\,\mathrm{d}t$. The metric tensor (3.9) allows us to define a notion of Riemannian gradients for functionals on $\mathcal{P}(\mathcal{Z})$. In the next section we shall compute this gradient explicitly for the squared MMD gradient flow.

4 Gradient Flow for Maximum Mean Discrepancy

As $\mathcal{P}(\mathcal{Z})$ is an infinite dimensional and curved space, many machine learning methods based on finite dimensional or linear structure cannot be directly applied to this manifold. To circumvent this problem, we use a positive definite kernel to map $\mathcal{P}(\mathcal{Z})$ to a RKHS and then perform our analysis on it. Let k be a positive definite kernel on $\mathcal{Z}$, and let $\mathcal{H}$ be the RKHS generated by k. The inner product on $\mathcal{H}$ is denoted by $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, and the kernel mean embedding $\rho \in \mathcal{P}(\mathcal{Z}) \longmapsto \mathbf{m}_\rho(\cdot) \in \mathcal{H}$ is given by $\mathbf{m}_\rho(z) := \int_{\mathcal{Z}} k(z, w)\, \rho(\mathrm{d}w)$ for z in $\mathcal{Z}$. The maximum mean discrepancy (MMD) (Gretton et al., 2012) between $\rho \in \mathcal{P}(\mathcal{Z})$ and the target ϱ is defined as the maximum, over all test functions in the unit ball of $\mathcal{H}$, of the mean difference between the two distributions. Moreover, it can be expressed by $\mathrm{MMD}(\rho, \varrho) = \|\mathbf{m}_\rho - \mathbf{m}_\varrho\|_{\mathcal{H}}$ (see Appendix A). When k is characteristic, the kernel mean embedding $\rho \mapsto \mathbf{m}_\rho$ is injective and therefore, $\mathrm{MMD}(\rho, \varrho) = 0$ if and only if $\rho = \varrho$.

Consider the loss function $\mathcal{F}[\rho] := \frac{1}{2}\mathrm{MMD}(\rho, \varrho)^2 = \frac{1}{2}\|\mathbf{m}_\rho - \mathbf{m}_\varrho\|^2_{\mathcal{H}}$. For each ρ, the Riemannian gradient $\operatorname{grad}\mathcal{F}[\rho]$ is defined as the unique element in $\mathrm{T}_\rho\mathcal{P}(\mathcal{Z})$ satisfying

$$g_\rho(\operatorname{grad}\mathcal{F}[\rho], \zeta) = \frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t=0}\mathcal{F}[\rho_t]$$

for every differentiable curve $t \mapsto \rho_t \in \mathcal{P}(\mathcal{Z})$ passing through ρ at $t = 0$ with tangent vector $\partial_t\rho_t|_{t=0} = \zeta$. By using the Riemannian metric tensor (3.9), we can compute explicitly this gradient.

Lemma 4.1 (Gradient formula)**.** *The Riemannian gradient of the functional $\mathcal{F}$ satisfies*

$$\operatorname{grad}\mathcal{F}[\rho] = -\mathrm{div}_d\left(\rho\nabla_d[\mathbf{m}_\rho - \mathbf{m}_\varrho]\right).$$

The Riemannian gradient $\operatorname{grad}\mathcal{F}$ on $\mathcal{P}(\mathcal{Z})$ depends not only on the gradient operator ∇_d but also on the divergence operator. Using Lemma 4.1, we can rewrite the gradient flow equation $\partial_t\rho_t = -\operatorname{grad}\mathcal{F}[\rho_t]$ explicitly as

$$\partial_t\rho_t = \mathrm{div}_d\big(\rho_t\nabla_d[\mathbf{m}_{\rho_t} - \mathbf{m}_\varrho]\big) \quad \text{for} \quad t \geq 0. \tag{4.1}$$

The next result exhibits the rate at which $\mathcal{F}$ decreases its value along the flow.

Proposition 4.2 (Rate of decrease)**.** *Along the gradient flow $t \mapsto \rho_t \in \mathcal{P}(\mathcal{Z})$ given by* (4.1)*, we have*

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}[\rho_t] = -\int_{\mathcal{Z}} \big\|\nabla_d[\mathbf{m}_{\rho_t} - \mathbf{m}_\varrho]\big\|_z^2\, \rho_t(\mathrm{d}z) \quad \textit{for} \quad t \geq 0.$$

Proposition 4.2 implies that $\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}[\rho_t] = 0$ if and only if $\nabla_d[\mathbf{m}_{\rho_t} - \mathbf{m}_\varrho](z) = 0$ for every z in the support of the distribution ρ_t. As a consequence, the objective function will decrease its value whenever the gradient $\nabla_d[\mathbf{m}_{\rho_t} - \mathbf{m}_\varrho]$ is not identically zero.

4.1 Riemannian Forward Euler Scheme

We now propose the Riemannian version of the forward Euler scheme to discretize the continuous flow (4.1):

$$\boxed{\rho^{\tau+1} = \exp(s_\tau\Phi^\tau)_{\#}\rho^\tau \quad \text{with} \quad \Phi^\tau := -\nabla_d[\mathbf{m}_{\rho^\tau} - \mathbf{m}_\varrho], \qquad (4.2)}$$

where $s_\tau > 0$ is the step size. Here, for a vector field $\Phi = (\Phi_1, \Phi_2, \Phi_3) : \mathcal{Z} \to \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{S}^n$ and for $\varepsilon \geq 0$, $\exp(\varepsilon\Phi) : \mathcal{Z} \to \mathcal{Z}$ is the Riemannian exponential map induced by (3.5), i.e.,

$$\exp_z(\varepsilon\Phi(z)) = \Big(x + \varepsilon\Phi_1(z),\ \mu + \varepsilon\Phi_2(z),\ (I + \varepsilon\mathrm{L}_\Sigma[\Phi_3(z)])\Sigma(I + \varepsilon\mathrm{L}_\Sigma[\Phi_3(z)])\Big)$$

for $z = (x, \mu, \Sigma) \in \mathcal{Z}$. Notice in the above equation that the input z affects simultaneously the bases of the exponential map $\exp_z$ as well as the direction $\Phi(z)$. This map is the ε-perturbation of the identity map along geodesics with directions Φ. When $\rho^\tau = N^{-1}\sum_{i=1}^N \delta_{z_i^\tau}$ is an empirical distribution that is supported on $(z_i^\tau)_{i=1}^N$, scheme (4.2) flows each particle z_i^τ to the new position $z_i^{\tau+1} = \exp_{z_i^\tau}(s_\tau\Phi(z_i^\tau))$. The next lemma shows that Φ^τ is the steepest descent direction for $\mathcal{F}$ w.r.t. the exponential map among all directions in the space $\mathbb{L}^2(\rho^\tau)$, which is the collection of all vector fields Φ on $\mathcal{Z}$ satisfying $\|\Phi\|^2_{\mathbb{L}^2(\rho^\tau)} := \int_{\mathcal{Z}} \|\Phi(z)\|_z^2\rho^\tau(\mathrm{d}z) < \infty$.

Algorithm 1 Discretized Gradient Flow Algorithm for Scheme (4.2)

```
Input: a source distribution $\rho^0 = N^{-1}\sum_{i=1}^N \delta_{z_i^0}$, a target distribution $\varrho = M^{-1}\sum_{j=1}^M \delta_{\bar{z}_j}$, a
number of iterations $T$, a sequence of step sizes $s_\tau > 0$ with $\tau = 0, 1, ..., T$ and a kernel $k$
Initialization: Compute $\bar{\Psi}(z) = M^{-1}\sum_{j=1}^M \nabla_d^1 k(z, \bar{z}_j)$ with $\nabla_d^1 k(z, \bar{z}_j)$ is $\nabla_d$ of $z \mapsto k(z, \bar{z}_j)$
repeat for each $\tau = 0, \dots, T-1$:
    Compute $\Psi^\tau(z) = N^{-1}\sum_{i=1}^N \nabla_d^1 k(z, z_i^\tau)$
    for $i = 1, \dots, N$ do $z_i^{\tau+1} \leftarrow \exp_{z_i^\tau}\big(s_\tau(\bar{\Psi} - \Psi^\tau)(z_i^\tau)\big)$ end for
Output: $\rho^T = N^{-1}\sum_{i=1}^N \delta_{z_i^T}$
```

Lemma 4.3 (Steepest descent direction)**.** *Fix a distribution $\rho^\tau \in \mathcal{P}(\mathcal{Z})$. For any vector field $\Phi : \mathcal{Z} \to \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{S}^n$, we have*

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0} \mathcal{F}[\exp(\varepsilon\Phi)_{\#}\rho^\tau] = \int_{\mathcal{Z}} \langle \nabla_d[\mathbf{m}_{\rho^\tau} - \mathbf{m}_\varrho](z), \Phi(z)\rangle_z \, \rho^\tau(\mathrm{d}z).$$

As a consequence, if we let $\hat{\Phi}^\tau$ be the unit vector field (w.r.t. $\|\cdot\|_{\mathbb{L}^2(\rho^\tau)}$ norm) in the direction of Φ^τ given in (4.2)*, then $\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\big|_{\varepsilon=0}\mathcal{F}[\exp(\varepsilon\hat{\Phi}^\tau)_{\#}\rho^\tau] = -\|\nabla_d[\mathbf{m}_{\rho^\tau} - \mathbf{m}_\varrho]\|_{\mathbb{L}^2(\rho^\tau)}$ and this is the fastest decay rate among all unit directions Φ in $\mathbb{L}^2(\rho^\tau)$.*

It follows from Lemma 4.3 that the discrete scheme (4.2) satisfies the Riemannian gradient descent property: if $\nabla_d[\mathbf{m}_{\rho^\tau} - \mathbf{m}_\varrho]$ is nonzero and if $s_\tau > 0$ is chosen sufficiently small, then $\mathcal{F}[\rho^{\tau+1}] < \mathcal{F}[\rho^\tau]$. In the Appendix (Proposition A.5), we quantify the amount of decrease of $\mathcal{F}$ at each iteration. An iterative algorithm that implements the flow (4.2) is described in Algorithm 1.

Complexity. For each iteration τ in Algorithm 1, its complexity is $O(N(Nm + n^3))$ where m is the feature's dimension, n is the reduced dimension ($n \ll m$), N is the number of particles.

Convergence guarantees. We now study the (weak) convergence of the solution ρ_t of the continuous gradient flow (4.1), as well as the discretized counterpart ρ^τ of flow (4.2), to the target distribution ϱ. When the kernel k is characteristic, this convergence is equivalent to $\lim_{t\to\infty} \mathrm{MMD}(\rho_t, \varrho) = 0$. Because the objective function $\mathcal{F}$ is not displacement convex (Arbel et al., 2019, Section 3.1), the convergent theory for gradient flows in (Ambrosio et al., 2008) does not apply even in the case of Euclidean spaces. In general, there is a possibility that $\mathrm{MMD}(\rho_t, \varrho)$ does not decrease to zero as t tends to infinity. In view of Proposition 4.2, this happens if the solutions ρ_t are trapped inside the set $\big\{\rho : \int_{\mathcal{Z}} \big\|\nabla_d[\mathbf{m}_\rho - \mathbf{m}_\varrho]\big\|_z^2 \rho(\mathrm{d}z) = 0\big\}$. For each distribution ρ on $\mathcal{Z}$, we define in Appendix A.2 a symmetric linear and positive operator $\mathbb{K}_\rho : \mathcal{H} \to \mathcal{H}$ having the property that $\langle \mathbb{K}_\rho[\mathbf{m}_\rho - \mathbf{m}_\varrho], \mathbf{m}_\rho - \mathbf{m}_\varrho\rangle_{\mathcal{H}} = \int_{\mathcal{Z}} \big\|\nabla_d[\mathbf{m}_\rho - \mathbf{m}_\varrho]\big\|_z^2 \rho(\mathrm{d}z)$ (see Lemma A.6 in the Appendix). We further shows in Proposition A.8 that ρ_t globally converges in MMD if the minimum eigenvalue λ_t of the operator $\mathbb{K}_{\rho_t}$ satisfies an integrability condition.

4.2 Noisy Riemannian Forward Euler Scheme

The analysis in Section 4.1 reveals that the gradient flows suffer from convergence issues if the residual $\mathbf{m}_{\rho_t} - \mathbf{m}_\varrho$ belongs to the null space of the operator $\mathbb{K}_{\rho_t}$. To resolve this, we employ graduated optimization (Arbel et al., 2019; Chaudhari et al., 2017; Gulcehre et al., 2016; 2017; Hazan et al., 2016) used for non-convex optimization in Euclidean spaces. Specifically, we modify algorithm (4.2) by injecting Gaussian noise into the exponential map at each iteration τ to obtain

$$\boxed{\rho^{\tau+1} = \exp(s_\tau\Phi^\tau)_{\#}\rho^{\tau,\beta_\tau} \quad \text{with } f^{\beta_\tau} : (z, u) \mapsto \exp_z(\beta_\tau u),\ \rho^{\tau,\beta_\tau} := f^{\beta_\tau}{}_{\#}(\rho^\tau \otimes g). \quad (4.3)}$$

Here g is a Gaussian measure with distribution $\mathcal{N}_{\mathbb{R}^m}(0,1) \otimes \mathcal{N}_{\mathbb{R}^n}(0,1) \otimes \mathcal{N}_{\mathbb{S}^n}(0,1)$ on the tangent space and $\mathcal{N}_{\mathbb{S}^n}(0,1)$ denotes an n-by-n symmetric matrix whose upper triangular elements are i.i.d. standard Gaussian random variables. When $\rho^\tau = N^{-1}\sum_{i=1}^N \delta_{z_i^\tau}$, scheme (4.3) flows each particle z_i^τ first to $z_i^{\tau,\beta_\tau} := \exp_{z_i^\tau}(\beta_\tau U)$ with noise $U \sim g$ and then to $z_i^{\tau+1} = \exp_{z_i^{\tau,\beta_\tau}}(s_\tau\Phi(z_i^{\tau,\beta_\tau}))$. Our next result extends Proposition 8 in (Arbel et al., 2019) for the standard quadratic cost on the Euclidean space to the nonstandard cost function d^2 on the *curved* Riemannian manifold $\mathcal{Z}_{++}$. It demonstrates that scheme (4.3) achieves the global minimum of $\mathcal{F}$ provided that k is a Lipschitz-gradient kernel and both the noise level β_τ and the step size s_τ are well controlled. The proof of Proposition 4.4 is given in Appendix A.2 and relies on arguments that are different from that of (Arbel et al., 2019).

Proposition 4.4 (Objective value decay for noisy scheme)**.** *Suppose that k is a Lipschitz-gradient kernel*[1] *with constant L, and the noise level β_τ satisfies*

$$\lambda \beta_\tau^2 \mathcal{F}[\rho^\tau] \le \int_{\mathcal{Z}} \|\Phi^\tau(z)\|_z^2 \, \rho^{\tau,\beta_\tau}(\mathrm{d}z) \tag{4.4}$$

for some constant $\lambda > 0$. Then for $\rho^{\tau+1}$ obtained from scheme (4.3)*, we have*

$$\mathcal{F}[\rho^{\tau+1}] \le \mathcal{F}[\rho^0] \, \exp\Big(-\lambda \sum\nolimits_{i=0}^{\tau} [s_i\big(1 - 2Ls_i\big)\beta_i^2]\Big).$$

In particular, $\mathcal{F}[\rho^\tau]$ tends to zero if the sequence $\sum_{i=0}^{\tau} s_i\big(1 - 2Ls_i\big)\beta_i^2$ goes to positive infinity. For an adaptive step size $s_\tau \le 1/4L$, this condition is met if, for example, β_τ is chosen of the form $(\tau s_\tau)^{-\frac{1}{2}}$ while still satisfying (4.4). The noise perturbs the direction of descent, whereas the step size determines how far to move along this perturbed direction. The noise level needs to be adjusted so that the gradient is not too blurred, but it does not necessarily decrease at each iteration. When the incumbent distribution ρ^τ is close to a local optimum, it is helpful to increase the noise level to escape the local optimum. We demonstrate in Lemma A.5 of the Appendix that any positive definite kernel k with bounded Hessian w.r.t. distance d is a Lipschitz-gradient kernel. On the other hand, the detailed algorithm of scheme (4.3) are provided in the Appendix B.

5 Numerical Experiments

We evaluate the proposed gradient flow on real-world datasets and then illustrate its application to augment samples for dataset of interest in transfer learning where only a few samples in the dataset of interest are available.

We consider four datasets: the MNIST (M), Fashion-MNIST (F) and the Kanji-MNIST (K) datasets, along with the USPS (U) dataset. All images are resized to 20 × 20, thus the feature space is of dimension $m = 400$. To satisfy the Gaussianity assumption of the conditional distributions, we apply K-means clustering to each dataset, and subsampling a smaller dataset using the biggest cluster.

We use t-distributed stochastic neighbor embedding (tSNE) as our mapping ϕ from $\mathbb{R}^m$ to $\mathbb{R}^2$. To compute the MMD distance using kernel embeddings, we use a tensor kernel on $\mathcal{Z}$ composed from three standard Gaussian kernels corresponding for each component of the feature space $\mathbb{R}^{400}$, the mean space $\mathbb{R}^2$ and the covariance matrix space $\mathbb{S}_{++}^2$. The kernel k is thus characteristic by (Szabó & Sriperumbudur, 2018, Theorem 4). Our algorithms and experiments are implemented in PyTorch. All the experiments are run on a machine with a NVIDIA Tesla V100 GPU and an Intel Xeon E5-2690 6-core v4 CPU. Codes and data are available in the supplementary file.

5.1 Flows between Datasets

In the first set of experiments, we examine the path travelled by each particle from the source domain to the target domain. To this end, we fix a source-domain pair, then we sample randomly $N = 200$ images equally for 10 classes of the source domain, and $M = 50$ images equally for 10 classes of the target domain. The results of our flows are depicted in Fig. 2. In each subfigure, each column represents a snapshot of a certain time-step and the samples flow from the source (left) to the target (right). The number of iterations T that is used to generate the results in Fig. 2 is capped at 140.

5.2 Transfer Learning

One application of the gradient flow approach is to alleviate the problem of insufficient labeled data by augmenting the target dataset with new samples. In this experiment, we demonstrate how new target domain samples obtained from our gradient flows can be used in a transfer learning setting.

[1] See Definition A.3 in the Appendix for the technical definition of a Lipschitz-gradient kernel

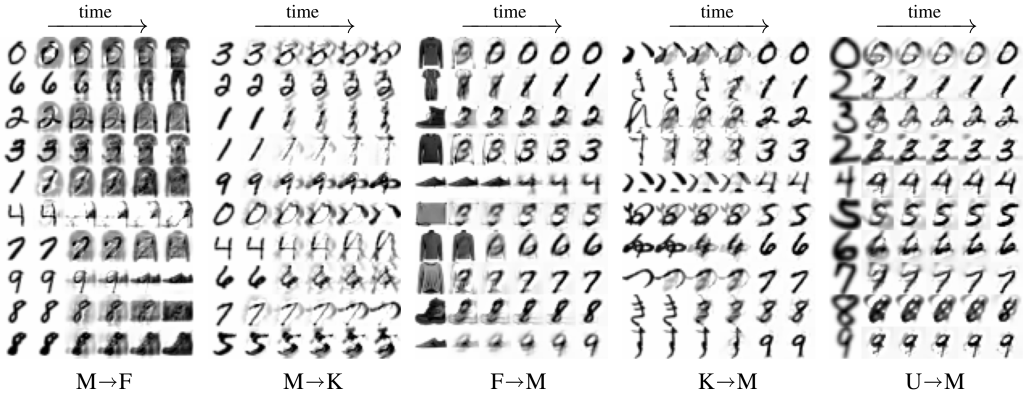

Figure 2: Sample path visualizations for different source-target domain combinations.

To this end, we fix a source domain, and pretrain a classifier P on this domain. This classifier is using LeNet-5's architecture. We also draw randomly $N = 200$ samples from the source domain (equal size for each class) to form the source dataset $(x_i, y_i)_{i=1}^{200}$. Next, we pick a target domain and draw randomly a few samples from this target domain: in 1-shot (resp. 5-shot) learning, only 1 image (resp. 5 images) per class from the target domain is drawn to form the target dataset $D = (\bar{x}_j, \bar{y}_j)_{j=1}^{M}$. We then perform a noisy gradient flow scheme (4.3) from the source dataset to the target dataset to get 200 new samples $S_T = (x_i^T, y_i^T)_{i=1}^{200}$. With the target dataset D and new samples S_T, we can retrain the classifier P with 10 epochs with Adam optimizer and learning rate 2×10^{-3}. Similarly, we can also train new networks from scratch using only D and S_T. Finally, we test the classifiers on the test set in the target domain. This process is replicated independently 10 times. We include more details on implementation in Section B.5.

In Fig. 3, we present the accuracy of different transfer learning strategies using the new labelled samples. D and $D \cup S_T$ mean training a new classifier from scratch, whereas P means transferring from the pretrained classifier. We observe a common trend that the addition of the new samples S_T always improves the accuracy of the classifiers. Both the 5-shot learning and 1-shot learning results demonstrate similar relative order of accuracy among approaches. Moreover, the data augmentation with S_T leads to higher increase of accuracy for the 1-shot learning. We compare with Alvarez-Melis & Fusi (2021) in transfer learning results and computation cost, see Section B.6.

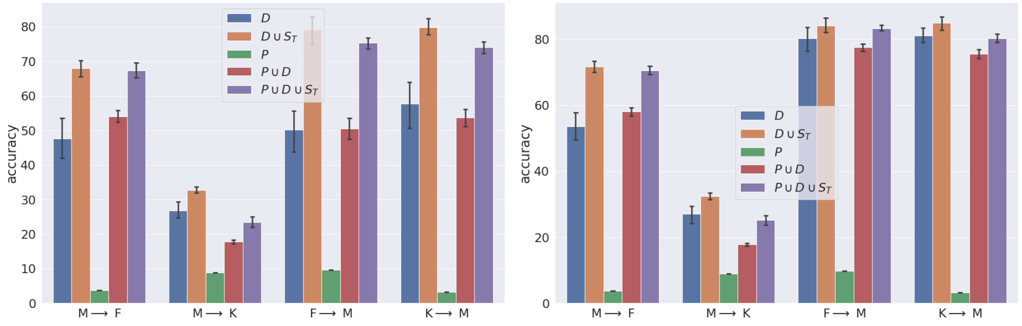

Figure 3: Average target domain accuracy and error bars for transfer learning with one-shot (left) and five-shot (right). Results are taken over 10 independent replications.

Concluding Remarks. This paper focuses on a gradient flow approach to generate new labelled data samples in the target domain. To overcome the discrete nature of the label set, we represent datasets as distributions on the feature-Gaussian space, and the flow is formulated to minimize an MMD loss function under an optimal transport metric. Contrary to existing gradient flows on linear structure, our flows are developed on the *curved* Riemannian manifold of Gaussian distributions. We provide explicit formula for the (Riemannian) gradient of the MMD loss function, and examine in details the flow equations and the convergence properties of both continuous and (noisy) discretized forms. The numerical experiments demonstrate that our method can generate sensible labelled training data for the target domain, and improve the classification accuracy in few-shot learning.

References

David Alvarez-Melis and Nicolo Fusi. Geometric dataset distances via optimal transport. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21428–21439, 2020.

David Alvarez-Melis and Nicolò Fusi. Dataset dynamics via gradient flows in probability space. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 219–230, 2021.

L. Ambrosio, N. Gigli, and G. Savaré. *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Birkhäuser Verlag, 2008.

M Arbel, A Gretton, W Li, and G Montufar. Kernelized Wasserstein natural gradient. In *International Conference on Learning Representations*, 2020.

Michael Arbel, Anna Korba, Adil Salim, and Arthur Gretton. Maximum mean discrepancy gradient flow. In *Advances in Neural Information Processing Systems*, volume 32, pp. 6481–6491, 2019.

Marc G. Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Remi Munos. The Cramer distance as a solution to biased Wasserstein gradients. In *arXiv:1705.10743*, 2017.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, volume 19, 2007.

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

R. Bhatia, T. Jain, and Y. Lim. On the Bures-Wasserstein distance between positive definite matrices. *Expositiones Mathematicae*, 37(2):165–191, 2019.

Rajendra Bhatia. *Matrix Analysis*. Springer, 1997.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018.

Leon Bottou, Martin Arjovsky, David Lopez-Paz, and Maxime Oquab. Geometrical insights for implicit generative modeling. In *In Braverman Readings in Machine Learning*, pp. 229–268, 2017.

Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor Sampedro, Kurt Konolige, Sergey Levine, and Vincent Vanhoucke. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. 2018. URL `https://arxiv.org/abs/1709.07857`.

Pratik Chaudhari, Adam M. Oberman, Stanley J. Osher, Stefano Soatto, and Guillaume Carlier. Deep relaxation: partial differential equations for optimizing deep neural networks. *CoRR*, 2017.

Yifan Chen and Wuchen Li. Optimal transport natural gradient for statistical manifolds with continuous sample space. In *Information Geometry*, 2018.

Lénaïc Chizat. Sparse optimization on measures with over-parameterized gradient descent. In *arXiv:1907.10300*, 2020.

Lénaïc Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. *CoRR*, abs/1803.10081, 2018.

A. Duncan, Nikolas Nüsken, and Lukasz Szpruch. On the geometry of Stein variational gradient descent. *CoRR*, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1126–1135, 2017.

Charlie Frogner and Tomaso Poggio. Approximate inference with Wasserstein gradient flows. In *International Conference on Artificial Intelligence and Statistics*, pp. 2581–2590. PMLR, 2020.

M. Gelbrich. On a formula for the L^2 Wasserstein metric between measures on Euclidean and Hilbert spaces. *Mathematische Nachrichten*, 147(1):185–203, 1990.

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2066–2073, 2012. doi: 10.1109/CVPR.2012.6247911.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

Caglar Gulcehre, Marcin Moczulski, Misha Denil, and Yoshua Bengio. Noisy activation functions. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 3059–3068, 2016.

Caglar Gulcehre, Marcin Moczulski, Francesco Visin, and Yoshua Bengio. Mollifying networks. In *5th International Conference on Learning Representations*, 2017.

Elad Hazan, Kfir Yehuda Levy, and Shai Shalev-Shwartz. On graduated optimization for stochastic non-convex problems. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1833–1841, 2016.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.

R. Jordan, D. Kinderlehrer, and F. Otto. The variational formulation of the Fokker–Planck equation. *SIAM Journal on Mathematical Analysis*, 29:1–17, 1998.

Mikhail Khodak, Maria-Florina F Balcan, and Ameet S Talwalkar. Adaptive gradient-based meta-learning methods. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, and Shahin Shahrampour. Generalized sliced distances for probability distributions. *arXiv prepritn arXiv:2002.12537*, 2020.

John Lee. *Introduction to Smooth Manifolds*. Springer-Verlag, 2003.

Qiang Liu. Stein variational gradient descent as gradient flow. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Antoine Liutkus, Umut Simsekli, Szymon Majewski, Alain Durmus, and Fabian-Robert Stöter. Sliced-Wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 4104–4113, 2019.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 2208–2217, 2017.

L. Malagò, L. Montrucchio, and G. Pistone. Wasserstein Riemannian geometry of Gaussian densities. *Information Geometry*, 1(2):137–179, 2018.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems*, volume 21, 2009.

Youssef Mroueh and Truyen Nguyen. On the convergence of gradient descent in GANs: MMD GAN as a gradient flow. In *International Conference on Artificial Intelligence and Statistics*, 2021.

Youssef Mroueh, Tom Sercu, and Anant Raj. Sobolev descent. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pp. 2976–2985, 2019.

F. Otto. The geometry of dissipative evolution equations: the porous medium equation. *Comm. Partial Differential Equations*, 26:101–174, 2001.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191.

Filippo Santambrogio. *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs and Modeling*. Birkhäuser, 2015.

Filippo Santambrogio. Euclidean, metric, and Wasserstein gradient flows: An overview. *Bullentin of Mathematical Sciences*, 7:87–154, 2017.

Zoltán Szabó and Bharath K. Sriperumbudur. Characteristic and universal tensor product kernels. *Journal of Machine Learning Research*, 18(233):1–29, 2018.

Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

Asuka Takatsu. Wasserstein geometry of Gaussian measures. *Osaka Journal of Mathematics*, 48(4): 1005–1026, 2011.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

C. Villani. *Optimal Transport: Old and New*. Springer Science & Business Media, 2008.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Amir R. Zamir, Alexander Sax, William Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

The Appendix is organized into two parts. In Section A, we provide the proofs and further discussions of the results in the main paper. Section B includes implementation details as well as additional numerical results. All the models and data used to create the results in the paper are in the supplementary file.

A Proofs

A.1 Proofs and Results related to Section 3

For Proposition 3.1. Recall that the Bures distance defined on $\mathbb{S}^n_{++}$ is

$$\mathbb{B}(\Sigma_1, \Sigma_2) := \big[\mathrm{tr}(\Sigma_1 + \Sigma_2 - 2[\Sigma_1^{\frac{1}{2}} \Sigma_2 \Sigma_1^{\frac{1}{2}}]^{\frac{1}{2}})\big]^{\frac{1}{2}}, \tag{A.1}$$

and $\nabla_d \varphi(z)$ is the unique element in the tangent space $\mathbb{R}^m \times \mathbb{R}^n \times \mathbb{S}^n$ satisfying

$$\big\langle \nabla_d \varphi(z), (w, v, V) \big\rangle_z = D\varphi_z(w, v, V) \quad \text{for all } (w, v, V) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{S}^n. \tag{A.2}$$

The proof of Proposition 3.1 relies on the following result from (Takatsu, 2011, Proposition A) (see also (Bhatia et al., 2019, Theorem 5) and (Malagò et al., 2018, Proposition 6)).

Proposition A.1. *The space $\mathbb{S}^n_{++}$ is a Riemannian manifold with the following structures: at each point $\Sigma \in \mathbb{S}^n_{++}$, the tangent space is $\mathrm{T}_\Sigma \mathbb{S}^n_{++} = \mathbb{S}^n$ and the Riemannian metric is given by*

$$\langle X, Y \rangle_\Sigma := tr\Big(\mathrm{L}_\Sigma[X]\, \Sigma\, \mathrm{L}_\Sigma[Y]\Big) = \frac{1}{2} \langle \mathrm{L}_\Sigma[X], Y \rangle \quad for \quad X, Y \in \mathbb{S}^n.$$

Moreover, the distance function corresponding to this Riemannian metric coincides with the Bures distance $\mathbb{B}$ given by (A.1).

We are now ready to prove Proposition 3.1.

Proof of Proposition 3.1. As a consequence of Proposition A.1, $\mathcal{Z}$ is the product Riemannian manifold with tangent space $\mathrm{T}_{(x,\mu,\Sigma)}\mathcal{Z} = \mathrm{T}_x\mathbb{R}^m \times \mathrm{T}_\mu \mathbb{R}^n \times \mathrm{T}_\Sigma \mathbb{S}^n_{++}$ and with the standard product Riemmanian metric (3.3). The result then follows. We note that if we let $\mathbb{D}((x_1, \mu_1, \Sigma_1), (x_2, \mu_2, \Sigma_2))$ denote the distance function corresponding to the Riemannian metric $\langle \cdot, \cdot \rangle_z$ on $\mathcal{Z}$, then its square $\mathbb{D}((x_1, \mu_1, \Sigma_1), (x_2, \mu_2, \Sigma_2))^2$ is the sum of the square of the distance function w.r.t. standard metric $\langle \cdot, \cdot \rangle$ on $\mathbb{R}^m$, the square of the distance function w.r.t. standard metric $\langle \cdot, \cdot \rangle$ on $\mathbb{R}^n$, and the square of the distance function w.r.t. metric $\langle \cdot, \cdot \rangle_\Sigma$ on $\mathbb{S}^n_{++}$. As a result and by Proposition A.1, we have $\mathbb{D}((x_1, \mu_1, \Sigma_1), (x_2, \mu_2, \Sigma_2))^2 = \|x_1 - x_2\|_2^2 + \|\mu_1 - \mu_2\|_2^2 + \mathbb{B}(\Sigma_1, \Sigma_2)^2$. So, $\mathbb{D}$ is the same as d. □

For Lemma 3.2

Proof of Lemma 3.2. Let us express $\nabla_d \varphi(z) = (\Phi_1(z), \Phi_2(z), \Phi_3(z))$ with $\Phi_1(z) \in \mathbb{R}^m$, $\Phi_2(z) \in \mathbb{R}^n$ and $\Phi_3(z) \in \mathbb{S}^n$. Then by using the definition of Riemannian metric $\langle \cdot, \cdot \rangle_z$ in (3.3), we can rewrite equation (A.2) as

$$\langle \Phi_1(z), v \rangle + \langle \Phi_2(z), w \rangle + \langle \frac{1}{2} \mathrm{L}_\Sigma[\Phi_3(z)], V \rangle = \langle \nabla \varphi(z), (v, w, V) \rangle.$$

This is equivalent to

$$\langle \Phi_1(z), v \rangle + \langle \Phi_2(z), w \rangle + \langle \frac{1}{2} \mathrm{L}_\Sigma[\Phi_3(z)], V \rangle = \langle \nabla_x \varphi(z), v \rangle + \langle \nabla_\mu \varphi(z), w \rangle + \langle \nabla_\Sigma \varphi(z), V \rangle, \tag{A.3}$$

where $\nabla \varphi(z) = \Big(\nabla_x \varphi(z), \nabla_\mu \varphi(z), \nabla_\Sigma \varphi(z)\Big)$ denotes the standard Euclidean gradient. Equation (A.3) is obviously satisfied if $\Phi_1(z) = \nabla_x \varphi(z)$, $\Phi_2(z) = \nabla_\mu \varphi(z)$, and $\mathrm{L}_\Sigma[\Phi_3(z)] = 2\nabla_\Sigma \varphi(z)$. By the definition of operator L_Σ right after (3.2), the third identity is the same as $\Phi_3(z) = 2[\nabla_\Sigma \varphi(z)]\Sigma + 2\Sigma[\nabla_\Sigma \varphi(z)]$. Due to uniqueness of the gradient, we therefore infer that $\nabla_d \varphi(z)$ is given by the formula:

$$\nabla_d \varphi(z) = \Big(\nabla_x \varphi(z), \nabla_\mu \varphi(z), 2[\nabla_\Sigma \varphi(z)]\Sigma + 2\Sigma[\nabla_\Sigma \varphi(z)]\Big).$$

This completes the proof. □

In this paper, the optimal transport metric between any two distributions $\rho_0, \rho_1 \in \mathcal{P}(\mathcal{Z})$ is defined by

$$\mathbb{W}(\rho_0,\rho_1)^2 \coloneqq \inf_{\pi\in\Pi(\rho_0,\rho_1)} \iint_{\mathcal{Z}\times\mathcal{Z}} d(z_0,z_1)^2 \; \pi(\mathrm{d}z_0,\mathrm{d}z_1), \tag{A.4}$$

where $\Pi(\rho_0,\rho_1)$ is the set of all probability distributions on $\mathcal{Z}\times\mathcal{Z}$ whose marginals are ρ_0 and ρ_1, respectively.

A.2 Proofs and Results related to Section 4

The maximum mean discrepancy (MMD) between a distribution $\rho \in \mathcal{P}(\mathcal{Z})$ and the target distribution ϱ is defined as

$$\mathrm{MMD}(\rho,\varrho) \coloneqq \sup_{f\in\mathcal{H}:\|f\|_{\mathcal{H}}\le 1} \Big\{ \int_{\mathcal{Z}} f(z)\,\rho(\mathrm{d}z) - \int_{\mathcal{Z}} f(z)\,\varrho(\mathrm{d}z)\Big\}.$$

It is well-known that the MMD admits the following closed-form formula (Gretton et al., 2012, Lemmas 4 and 6).

Lemma A.2. *We have* $\mathrm{MMD}(\rho,\varrho) = \|\mathbf{m}_\rho - \mathbf{m}_\varrho\|_{\mathcal{H}}$. *As a consequence,*

$$\mathrm{MMD}(\rho,\varrho)^2 = \int_{\mathcal{Z}}\int_{\mathcal{Z}} k(z,w)\rho(\mathrm{d}z)\rho(\mathrm{d}w) - 2\int_{\mathcal{Z}} \mathbf{m}_\varrho(z)\rho(\mathrm{d}z) + \|\mathbf{m}_\varrho\|_{\mathcal{H}}^2.$$

Proof of Lemma A.2. For any $f \in \mathcal{H}$, we have $f(z) = \langle f, k(\cdot,z)\rangle_{\mathcal{H}}$. Therefore,

$$\int_{\mathcal{Z}} f(z)\,\rho(\mathrm{d}z) = \Big\langle f, \int_{\mathcal{Z}} k(\cdot,z)\,\rho(\mathrm{d}z)\Big\rangle_{\mathcal{H}} = \langle f, \mathbf{m}_\rho\rangle_{\mathcal{H}} \quad \text{for all} \quad f\in\mathcal{H}. \tag{A.5}$$

It follows that $\mathrm{MMD}(\rho,\varrho) = \sup_{f\in\mathcal{H}:\|f\|_{\mathcal{H}}\le1}\langle f, \mathbf{m}_\rho - \mathbf{m}_\varrho\rangle_{\mathcal{H}} = \|\mathbf{m}_\rho - \mathbf{m}_\varrho\|_{\mathcal{H}}$. Using this closed-form formula and identity (A.5), we also obtain

$$\begin{aligned}\mathrm{MMD}(\rho,\varrho)^2 &= \|\mathbf{m}_\rho - \mathbf{m}_\varrho\|_{\mathcal{H}}^2 = \langle \mathbf{m}_\rho, \mathbf{m}_\rho\rangle_{\mathcal{H}} - 2\langle \mathbf{m}_\rho, \mathbf{m}_\varrho\rangle_{\mathcal{H}} + \langle \mathbf{m}_\varrho, \mathbf{m}_\varrho\rangle_{\mathcal{H}}\\ &= \int \mathbf{m}_\rho(z)\rho(\mathrm{d}z) - 2\int \mathbf{m}_\varrho(z)\rho(\mathrm{d}z) + \|\mathbf{m}_\varrho\|_{\mathcal{H}}^2\\ &= \iint k(z,w)\rho(\mathrm{d}z)\rho(\mathrm{d}w) - 2\int \mathbf{m}_\varrho(z)\rho(\mathrm{d}z) + \|\mathbf{m}_\varrho\|_{\mathcal{H}}^2.\end{aligned}$$

This completes the proof. □

For Lemma 4.1

Proof of Lemma 4.1. We recall that $\operatorname{grad}\mathcal{F}[\rho]$ is defined as the unique element in $\mathrm{T}_\rho\mathcal{P}(\mathcal{Z})$ satisfying

$$g_\rho\Big(\operatorname{grad}\mathcal{F}[\rho], \partial_t\rho_t|_{t=0}\Big) = \frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t=0}\mathcal{F}[\rho_t]$$

for every differentiable curve $t\mapsto\rho_t\in\mathcal{P}(\mathcal{Z})$ passing through ρ at $t=0$. Let $t\mapsto\rho_t\in\mathcal{P}(\mathcal{Z})$ be such a curve. Then since $\partial_t\rho_t|_{t=0}\in\mathrm{T}_\rho\mathcal{P}(\mathcal{Z})$, we can write $\partial_t\rho_t|_{t=0} = -\mathrm{div}_d(\rho\nabla_d\varphi)$ for some differentiable function φ on $\mathcal{Z}$. Then by using Lemma A.2 and $k(z,w)=k(w,z)$ we have

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t=0}\mathcal{F}[\rho_t] &= \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t=0}\Big[\iint k(z,w)\rho_t(\mathrm{d}z)\rho_t(\mathrm{d}w) - 2\int \mathbf{m}_\varrho(z)\rho_t(\mathrm{d}z)\Big]\\ &= \frac{1}{2}\iint k(z,w)\partial_t\rho_t|_{t=0}(\mathrm{d}z)\rho(\mathrm{d}w) + \frac{1}{2}\iint k(z,w)\partial_t\rho_t|_{t=0}(\mathrm{d}w)\rho(\mathrm{d}z)\\ &\quad - \int \mathbf{m}_\varrho(z)\,\partial_t\rho_t|_{t=0}(\mathrm{d}z)\\ &= -\int_{\mathcal{Z}}\int_{\mathcal{Z}} k(z,w)\mathrm{div}_d(\rho\nabla_d\varphi)(\mathrm{d}z)\,\rho(\mathrm{d}w) + \int_{\mathcal{Z}} \mathbf{m}_\varrho(z)\,\mathrm{div}_d(\rho\nabla_d\varphi)(\mathrm{d}z).\end{aligned}$$

Let $\nabla_d^1 k(z,w)$ denote the gradient ∇_d of the function $z \mapsto k(z,w$. It then follows from the definition of the divergence operator $\mathrm{div}_d(\rho\nabla_d\varphi)$ at the end of Section 3.1 that

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t=0}\mathcal{F}[\rho_t] &= \int_{\mathcal{Z}}\int_{\mathcal{Z}}\langle\nabla_d^1 k(z,w),\nabla_d\varphi(z)\rangle_z\,\rho(\mathrm{d}z)\rho(\mathrm{d}w) - \int_{\mathcal{Z}}\langle\nabla_d\mathbf{m}_\varrho(z),\nabla_d\varphi(z)\rangle_z\rho(\mathrm{d}z)\\ &= \int\left[\Big\langle\int\nabla_d^1 k(z,w)\,\rho(\mathrm{d}w),\nabla_d\varphi(z)\Big\rangle_z\right]\rho(\mathrm{d}z) - \int_{\mathcal{Z}}\langle\nabla_d\mathbf{m}_\varrho(z),\nabla_d\varphi(z)\rangle_z\rho(\mathrm{d}z)\\ &= \int\left[\Big\langle\nabla_d\int k(z,w)\,\rho(\mathrm{d}w),\nabla_d\varphi(z)\Big\rangle_z\right]\rho(\mathrm{d}z) - \int_{\mathcal{Z}}\langle\nabla_d\mathbf{m}_\varrho(z),\nabla_d\varphi(z)\rangle_z\rho(\mathrm{d}z)\\ &= \int_{\mathcal{Z}}\langle\nabla_d[\mathbf{m}_\rho-\mathbf{m}_\varrho](z),\nabla_d\varphi(z)\rangle_z\rho(\mathrm{d}z).\end{aligned}$$

By the definition of the Riemannian metric tensor g_ρ given in (3.9) and due to $\partial_t\rho_t|_{t=0} = -\mathrm{div}_d(\rho\nabla_d\varphi)$, we thus obtain

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t=0}\mathcal{F}[\rho_t] &= g_\rho\Big(-\mathrm{div}_d\big(\rho\nabla_d[\mathbf{m}_\rho-\mathbf{m}_\varrho]\big), -\mathrm{div}_d(\rho\nabla_d\varphi)\Big)\\ &= g_\rho\Big(-\mathrm{div}_d\big(\rho\nabla_d[\mathbf{m}_\rho-\mathbf{m}_\varrho]\big), \partial_t\rho_t|_{t=0}\Big).\end{aligned}$$

Therefore, we infer that $\mathrm{grad}\,\mathcal{F}[\rho] = -\mathrm{div}_d\big(\rho\nabla_d[\mathbf{m}_\rho-\mathbf{m}_\varrho]\big)$ as desired. □

For Proposition 4.2

Proof of Proposition 4.2. The proof is similar to that of Lemma 4.1 and with the same notation for $\nabla_d^1 k(z,w)$. Indeed, by the same computation at the beginning of the proof of Lemma 4.1 we have

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}[\rho_t] &= \frac{1}{2}\iint k(z,w)\partial_t\rho_t(\mathrm{d}z)\rho_t(\mathrm{d}w) + \frac{1}{2}\iint k(z,w)\partial_t\rho_t(\mathrm{d}w)\rho_t(\mathrm{d}z) - \int\mathbf{m}_\varrho(z)\,\partial_t\rho_t(\mathrm{d}z)\\ &= \iint k(z,w)\partial_t\rho_t(\mathrm{d}z)\rho_t(\mathrm{d}w) - \int\mathbf{m}_\varrho(z)\,\partial_t\rho_t(\mathrm{d}z).\end{aligned}$$

This together with the gradient flow equation (4.1) gives

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}[\rho_t] &= \int_{\mathcal{Z}}\int_{\mathcal{Z}} k(z,w)\mathrm{div}_d\big(\rho_t\nabla_d[\mathbf{m}_{\rho_t}-\mathbf{m}_\varrho]\big)(\mathrm{d}z)\rho_t(\mathrm{d}w)\\ &\quad - \int_{\mathcal{Z}}\mathbf{m}_\varrho(z)\,\mathrm{div}_d\big(\rho_t\nabla_d[\mathbf{m}_{\rho_t}-\mathbf{m}_\varrho]\big)(\mathrm{d}z).\end{aligned}$$

Using the definition of the divergence operator div_d at the end of Section 3.1, we further obtain

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}[\rho_t] &= -\int_{\mathcal{Z}}\int_{\mathcal{Z}}\langle\nabla_d^1 k(z,w),\nabla_d[\mathbf{m}_{\rho_t}-\mathbf{m}_\varrho](z)\rangle_z\,\rho_t(\mathrm{d}z)\rho_t(\mathrm{d}w)\\ &\quad + \int_{\mathcal{Z}}\langle\nabla_d\mathbf{m}_\varrho(z),\nabla_d[\mathbf{m}_{\rho_t}-\mathbf{m}_\varrho](z)\rangle_z\rho_t(\mathrm{d}z)\\ &= -\int\left[\Big\langle\int\nabla_d^1 k(z,w)\,\rho_t(\mathrm{d}w),\nabla_d[\mathbf{m}_{\rho_t}-\mathbf{m}_\varrho](z)\Big\rangle_z\right]\rho_t(\mathrm{d}z)\\ &\quad + \int_{\mathcal{Z}}\langle\nabla_d\mathbf{m}_\varrho(z),\nabla_d[\mathbf{m}_{\rho_t}-\mathbf{m}_\varrho](z)\rangle_z\rho_t(\mathrm{d}z)\\ &= -\int_{\mathcal{Z}}\langle\nabla_d\mathbf{m}_{\rho_t}(z),\nabla_d[\mathbf{m}_{\rho_t}-\mathbf{m}_\varrho](z)\rangle_z\rho_t(\mathrm{d}z)\\ &\quad + \int_{\mathcal{Z}}\langle\nabla_d\mathbf{m}_\varrho(z),\nabla_d[\mathbf{m}_{\rho_t}-\mathbf{m}_\varrho](z)\rangle_z\rho_t(\mathrm{d}z)\\ &= -\int_{\mathcal{Z}}\|\nabla_d[\mathbf{m}_{\rho_t}-\mathbf{m}_\varrho](z)\|_z^2\rho_t(\mathrm{d}z).\end{aligned}$$

This yields the desired result. □

For Lemma 4.3

Proof of Lemma 4.3. From the formula for $\exp_z(\varepsilon\Phi(z))$ given at the beginning of Section 4.1, we observe that

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\exp_z(\varepsilon\Phi(z)) = \Big(\Phi_1(z), \Phi_2(z), \mathrm{L}_\Sigma[\Phi_3(z)]\,\Sigma + \Sigma\,\mathrm{L}_\Sigma[\Phi_3(z)]\Big)$$
$$= \Big(\Phi_1(z), \Phi_2(z), \Phi_3(z)\Big) = \Phi(z), \tag{A.6}$$

where the second equality is due to the definition of $\mathrm{L}_\Sigma[V]$ given at the beginning of Section 3.1.

We obtain from Lemma A.2 that

$$\mathrm{MMD}(\exp(\varepsilon\Phi)_\#\rho^\tau, \varrho)^2$$
$$= \iint k(z,w)\exp(\varepsilon\Phi)_\#\rho^\tau(\mathrm{d}z)\exp(\varepsilon\Phi)_\#\rho^\tau(\mathrm{d}w) - 2\int \mathbf{m}_\varrho(z)\exp(\varepsilon\Phi)_\#\rho^\tau(\mathrm{d}z) + \|\mathbf{m}_\varrho\|_{\mathcal{H}}^2$$
$$= \iint k\Big(\exp_z(\varepsilon\Phi(z)), \exp_w(\varepsilon\Phi(w))\Big)\rho^\tau(\mathrm{d}z)\rho^\tau(\mathrm{d}w) - 2\int \mathbf{m}_\varrho\Big(\exp_z(\varepsilon\Phi(z))\Big)\rho^\tau(\mathrm{d}z) + \|\mathbf{m}_\varrho\|_{\mathcal{H}}^2.$$

Moreover, we have

$$\iint \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\Big[k(\exp_z(\varepsilon\Phi(z)), \exp_w(\varepsilon\Phi(w)))\Big]\rho^\tau(\mathrm{d}z)\rho^\tau(\mathrm{d}w)$$
$$= \iint \left\{\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\Big[k(\exp_z(\varepsilon\Phi(z)), w)\Big] + \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\Big[k(z, \exp_w(\varepsilon\Phi(w)))\Big]\right\}\rho^\tau(\mathrm{d}z)\rho^\tau(\mathrm{d}w)$$
$$= \int \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\Big[\int k(\exp_z(\varepsilon\Phi(z)), w)\rho^\tau(\mathrm{d}w)\Big]\rho^\tau(\mathrm{d}z) + \int \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\Big[\int k(z, \exp_w(\varepsilon\Phi(w)))\rho^\tau(\mathrm{d}z)\Big]\rho^\tau(\mathrm{d}w)$$
$$= \int \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\Big[\mathbf{m}_{\rho^\tau}(\exp_z(\varepsilon\Phi(z)))\Big]\rho^\tau(\mathrm{d}z) + \int \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\Big[\mathbf{m}_{\rho^\tau}(\exp_w(\varepsilon\Phi(w)))\Big]\rho^\tau(\mathrm{d}w)$$
$$= 2\int \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\Big[\mathbf{m}_{\rho^\tau}(\exp_z(\varepsilon\Phi(z)))\Big]\rho^\tau(\mathrm{d}z).$$

Thus, it follows that

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\mathrm{MMD}(\exp(\varepsilon\Phi)_\#\rho^\tau, \varrho)^2$$
$$= 2\int \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\Big[\mathbf{m}_{\rho^\tau}(\exp_z(\varepsilon\Phi(z)))\Big]\rho^\tau(\mathrm{d}z) - 2\int \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\Big[\mathbf{m}_\varrho(\exp_z(\varepsilon\Phi(z)))\Big]\rho^\tau(\mathrm{d}z)$$
$$= 2\int D[\mathbf{m}_{\rho^\tau} - \mathbf{m}_\varrho]_z\Big(\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\exp_z(\varepsilon\Phi(z))\Big)\rho^\tau(\mathrm{d}z)$$

with $D\varphi_z(w,,v,V)$ denoting the standard directional derivative of φ at z in the direction (w,v,V). Using the definition of $\mathcal{F}$ together with (A.6) and the definition of gradient ∇_d in (A.2), we obtain

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\mathcal{F}[\exp(\varepsilon\Phi)_\#\rho^\tau] = \int D[\mathbf{m}_{\rho^\tau} - \mathbf{m}_\varrho]_z(\Phi(z))\,\rho^\tau(\mathrm{d}z)$$
$$= \int \langle\nabla_d[\mathbf{m}_{\rho^\tau} - \mathbf{m}_\varrho](z), \Phi(z)\rangle_z\rho^\tau(\mathrm{d}z). \tag{A.7}$$

This yields the first conclusion of the lemma.

Now let $\hat{\Phi}^\tau := \frac{\Phi^\tau}{\|\Phi^\tau\|_{\mathbb{L}^2(\rho^\tau)}}$ be the unit vector field in the direction of $\Phi^\tau := -\nabla_d[\mathbf{m}_{\rho^\tau} - \mathbf{m}_\varrho]$. Then by (A.7), we have

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\mathcal{F}[\exp(\varepsilon\hat{\Phi}^\tau)_\#\rho^\tau] = -\|\Phi^\tau\|_{\mathbb{L}^2(\rho^\tau)}^{-1}\int \|\Phi^\tau(z)\|_z^2\rho^\tau(\mathrm{d}z) = -\|\Phi^\tau\|_{\mathbb{L}^2(\rho^\tau)} \le 0.$$

On the other hand, for any unit direction Φ in $\mathbb{L}^2(\rho^\tau)$ we obtain from (A.7) and Hölder inequality that

$$\Big|\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\mathcal{F}[\exp(\varepsilon\Phi)_\#\rho^\tau]\Big| \le \int \|\Phi^\tau(z)\|_z\|\Phi(z)\|_z\rho^\tau(\mathrm{d}z)$$
$$\le \Big(\int \|\Phi^\tau(z)\|_z^2\rho^\tau(\mathrm{d}z)\Big)^{\frac{1}{2}}\Big(\int \|\Phi(z)\|_z^2\rho^\tau(\mathrm{d}z)\Big)^{\frac{1}{2}} = \|\Phi^\tau\|_{\mathbb{L}^2(\rho^\tau)}.$$

Therefore, we conclude further that

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\mathcal{F}[\exp(\varepsilon\hat{\Phi}^\tau)_\# \rho^\tau] \le \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\Big|_{\varepsilon=0}\mathcal{F}[\exp(\varepsilon\Phi)_\# \rho^\tau]$$

for any unit direction Φ in $\mathbb{L}^2(\rho^\tau)$. These give the last conclusion of the lemma. □

Definition A.3 (Lipschitz-gradient kernel)**.** *Let $L > 0$. A differentiable kernel k on $\mathcal{Z}$ is called a Lipschitz-gradient kernel with constant L if there exists a number $\varepsilon_0 \in (0,1)$ such that*

$$\begin{aligned}\Big|k(\exp_z(\varepsilon\Phi(z)), \exp_w(\delta\Phi(w))) - k(z,w) - \big[\langle \nabla_d^1 k(z,w), \varepsilon\Phi(z)\rangle_z + \langle \nabla_d^2 k(z,w), \delta\Phi(w)\rangle_w\big]\Big| \\ \le L\Big[\|\varepsilon\Phi(z)\|_z^2 + \|\delta\Phi(w)\|_w^2\Big]\end{aligned} \tag{A.8}$$

for every ε, $\delta \in [0,\varepsilon_0]$ and every bounded vector field $\Phi : \mathcal{Z} \to \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{S}^n$. Hereafter, $\nabla_d^1 k(z,w)$ and $\nabla_d^2 k(z,w)$ denote respectively the gradient ∇_d of the function $z \mapsto k(z,w)$ and the function $w \mapsto k(z,w)$.

Remark A.4. *The right hand side of condition* (A.8) *can be expressed in terms of the d distance as*

$$d\big(\exp_z(\varepsilon\Phi(z)), z\big)^2 + d\big(\exp_w(\delta\Phi(w)), w\big)^2.$$

Thus condition (A.8) *can be interpreted as the gradient $\nabla_d k$ is Lipschitz w.r.t. the distance d.*

Condition (A.8) is motivated by the following observation in the Euclidean space. Assume that $G : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a differentiable function such that its Euclidean gradient $\nabla G(z,w) := (\nabla^1 G(z,w), \nabla^2 G(z,w))$ satisfies the standard Lipschitz condition

$$\|\nabla G(z_1,w_1) - \nabla G(z_2,w_2)\|_2 \le L\|(z_1,w_1) - (z_2,w_2)\|_2 \quad \forall (z_1,w_1),\ (z_2,w_2) \in \mathbb{R}^d \times \mathbb{R}^d.$$

Then for any point $(z,w) \in \mathbb{R}^d \times \mathbb{R}^d$ and any tangent vector $(u,v) \in \mathbb{R}^d \times \mathbb{R}^d$, we have by using the mean value theorem that $G(z+u, w+v) - G(z,w) = \langle \nabla G(z_0,w_0), (u,v)\rangle$ for some point (z_0,w_0) in the line segment in $\mathbb{R}^d \times \mathbb{R}^d$ connecting the points (z,w) and $(z+u, w+v)$. As a consequence, we obtain

$$\begin{aligned}&\Big|G(z+u,w+v) - G(z,w) - \big[\langle \nabla^1 G(z,w), u\rangle + \langle \nabla^2 G(z,w), v\rangle\big]\Big| \\ &= \Big|\langle \nabla G(z_0,w_0), (u,v)\rangle - \langle \nabla G(z,w), (u,v)\rangle\Big| \\ &= \Big|\big\langle \nabla G(z_0,w_0) - \nabla G(z,w), (u,v)\big\rangle\Big| \le \|\nabla G(z_0,w_0) - \nabla G(z,w)\|_2\|(u,v)\|_2.\end{aligned}$$

Then we can use the Lipschitz condition for ∇G to imply further that

$$\begin{aligned}\Big|G(z+u,w+v) - G(z,w) - \big[\langle \nabla^1 G(z,w), u\rangle + \langle \nabla^2 G(z,w), v\rangle\big]\Big| \\ \le L\|(z_0,w_0) - (z,w)\|_2\|(u,v)\|_2 \le L\|(u,v)\|_2^2,\end{aligned}$$

which is the same as

$$\Big|G(z+u,w+v) - G(z,w) - \big[\langle \nabla^1 G(z,w), u\rangle + \langle \nabla^2 G(z,w), v\rangle\big]\Big| \le L\big[\|u\|_2^2 + \|v\|_2^2\big].$$

Condition (A.8) is the Riemannian version of this last inequality for the Euclidean space, which is a consequence of the standard Lipschitz condition for the gradient.

Bounded Hessian kernels are Lipschitz-gradient. The following lemma gives a sufficient condition for a kernel to be Lipschitz-gradient.

Lemma A.5. *Let k be a positive definite kernel such that its Hessian w.r.t. distance d is bounded. Then k is a Lipschitz-gradient kernel.*

Proof of Lemma A.5. Let $H_d^1 k(z,w)$ and $H_d^2 k(z,w)$ denote respectively the Hessian w.r.t. distance d of the function $z \mapsto k(z,w)$ and the function $w \mapsto k(z,w)$. Let ε, $\delta > 0$, and $\Phi : \mathcal{Z} \to \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{S}^n$

be a bounded vector field. Define $\gamma_z(t) := \exp_z(t\Phi(z))$ and $\theta_w(t) := \exp_w(\frac{\delta}{\varepsilon}t\Phi(w))$ for $t \in [0,\varepsilon]$. Then we have

$$k(\exp_z(\varepsilon\Phi(z)), \exp_w(\delta\Phi(w))) - k(z,w) = \int_0^\varepsilon \frac{\mathrm{d}}{\mathrm{d}t}[k(\gamma_z(t), \theta_w(t))]\,\mathrm{d}t$$
$$= \int_0^\varepsilon \Big[\langle \nabla_d^1 k(\gamma_z(t), \theta_w(t)), \dot\gamma_z(t)\rangle_{\gamma_z(t)} + \langle \nabla_d^2 k(\gamma_z(t), \theta_w(t)), \dot\theta_w(t)\rangle_{\theta_w(t)}\Big]\mathrm{d}t.$$

This together with the facts that $\dot\gamma_z(0) = \Phi(z)$ and $\dot\theta_w(0) = \frac{\delta}{\varepsilon}\Phi(z)$ yields

$$A := k(\exp_z(\varepsilon\Phi(z)), \exp_w(\delta\Phi(w))) - k(z,w) - \big[\langle \nabla_d^1 k(z,w), \varepsilon\Phi(z)\rangle_z + \langle \nabla_d^2 k(z,w), \delta\Phi(w)\rangle_w\big]$$
$$= \int_0^\varepsilon \Big[\langle \nabla_d^1 k(\gamma_z(t), \theta_w(t)), \dot\gamma_z(t)\rangle_{\gamma_z(t)} - \langle \nabla_d^1 k(\gamma_z(0), \theta_w(0)), \dot\gamma_z(0)\rangle_{\gamma_z(0)}\Big]dt$$
$$+ \int_0^\varepsilon \Big[\langle \nabla_d^2 k(\gamma_z(t), \theta_w(t)), \dot\theta_w(t)\rangle_{\theta_w(t)} - \langle \nabla_d^2 k(\gamma_z(0), \theta_w(0)), \dot\theta_w(0)\rangle_{\theta_w(0)}\Big]\mathrm{d}t$$
$$= \int_0^\varepsilon \int_0^t \frac{\mathrm{d}}{\mathrm{d}s}\Big[\langle \nabla_d^1 k(\gamma_z(s), \theta_w(s)), \dot\gamma_z(s)\rangle_{\gamma_z(s)}\Big]\mathrm{d}s\mathrm{d}t$$
$$+ \int_0^\varepsilon \int_0^t \frac{\mathrm{d}}{\mathrm{d}s}\Big[\langle \nabla_d^2 k(\gamma_z(s), \theta_w(s)), \dot\theta_w(s)\rangle_{\theta_w(s)}\Big]\mathrm{d}s\mathrm{d}t$$
$$= \int_0^\varepsilon \int_0^t \Big[\langle H_d^1 k(\gamma_z(s), \theta_w(s))\dot\gamma_z(s), \dot\gamma_z(s)\rangle_{\gamma_z(s)} + \langle \nabla_d^1 k(\gamma_z(s), \theta_w(s)), \ddot\gamma_z(s)\rangle_{\gamma_z(s)}\Big]\mathrm{d}s\mathrm{d}t$$
$$+ \int_0^\varepsilon \int_0^t \Big[\langle H_d^2 k(\gamma_z(s), \theta_w(s))\dot\theta_w(s), \dot\theta_w(s)\rangle_{\theta_w(s)} + \langle \nabla_d^2 k(\gamma_z(s), \theta_w(s)), \ddot\theta_w(s)\rangle_{\theta_w(s)}\Big]\mathrm{d}s\mathrm{d}t.$$

Since the curve $s \mapsto \gamma_z(s)$ is a geodesic, its acceleration $\ddot\gamma_z(s)$ is orthogonal to $\mathcal{Z}$ (that is, $\ddot\gamma_z(s)$ is orthogonal to every tangent vector in $\mathrm{T}_z\mathcal{Z}$). This implies that $\langle \nabla_d^1 k(\gamma_z(s), \theta_w(s)), \ddot\gamma_z(s)\rangle_{\gamma_z(s)} = 0$. Likewise, we also have $\langle \nabla_d^2 k(\gamma_z(s), \theta_w(s)), \ddot\theta_w(s)\rangle_{\theta_w(s)} = 0$. Thanks to these, we deduce from the above identity that

$$A = \int_0^\varepsilon \int_0^t \langle H_d^1 k(\gamma_z(s), \theta_w(s))\dot\gamma_z(s), \dot\gamma_z(s)\rangle_{\gamma_z(s)}\mathrm{d}s\mathrm{d}t$$
$$+ \int_0^\varepsilon \int_0^t \langle H_d^2 k(\gamma_z(s), \theta_w(s))\dot\theta_w(s), \dot\theta_w(s)\rangle_{\theta_w(s)}\mathrm{d}s\mathrm{d}t.$$

By using the assumption that the Hessians H_d^1 and H_d^2 are bounded, we then obtain

$$|A| \le M \int_0^\varepsilon \int_0^t \Big[\|\dot\gamma_z(s)\|^2_{\gamma_z(s)} + \|\dot\theta_w(s)\|^2_{\theta_w(s)}\Big]\mathrm{d}s\mathrm{d}t,$$

where M is the sup norm of the Hessian of k. But as $\gamma_z(s)$ and $\theta_w(s)$ are geodesic, they have constant speeds. Therefore, $\|\dot\gamma_z(s)\|_{\gamma_z(s)} = \|\dot\gamma_z(0)\|_{\gamma_z(0)} = \|\Phi(z)\|_z$ and $\|\dot\theta_w(s)\|_{\theta_w(s)} = \|\dot\theta_w(0)\|_{\theta_w(0)} = \|\frac{\delta}{\varepsilon}\Phi(z)\|_z$. Using these, we infer further that

$$|A| \le M \int_0^\varepsilon \int_0^t \Big[\|\Phi(z)\|_z^2 + (\frac{\delta}{\varepsilon})^2\|\Phi(z)\|_z^2\Big]\mathrm{d}s\mathrm{d}t = \frac{M}{2}\Big[\|\varepsilon\Phi(z)\|_z^2 + \|\delta\Phi(w)\|_w^2\Big].$$

According to Definition A.3, we thus conclude that k is a Lipschitz-gradient kernel with constant $M/2$. □

Quantified estimate of decrease for the Riemannian forward Euler scheme (4.2). The next result quantifies the amount that the value of $\mathcal{F}$ decreases after each iteration.

Proposition A.6 (Quantified estimate of decrease). *Suppose that k is a Lipschitz-gradient kernel with constant L. Then for $\rho^{\tau+1}$ given by* (4.2) *with $s_\tau \in (0, \varepsilon_0]$, we have*

$$\mathcal{F}[\rho^{\tau+1}] - \mathcal{F}[\rho^\tau] \le -s_\tau\big(1 - 2Ls_\tau\big)\int_{\mathcal{Z}} \|\nabla_d[\mathbf{m}_{\rho^\tau} - \mathbf{m}_\varrho](z)\|_z^2\,\rho^\tau(\mathrm{d}z).$$

Proof of Proposition A.6. Let $\Phi^\tau := -\nabla_d[\mathbf{m}_{\rho^\tau} - \mathbf{m}_\varrho]$. Then from the computation at the beginning of the proof of Lemma 4.3 and by using Lemma A.2, we obtain

$$\begin{aligned}\mathcal{F}[\rho^{\tau+1}] - \mathcal{F}[\rho^\tau] &= \frac{1}{2}\left[\mathrm{MMD}(\exp(s_\tau \Phi^\tau)_{\#}\rho^\tau, \varrho)^2 - \mathrm{MMD}(\rho^\tau, \varrho)^2\right] \\ &= \frac{1}{2}\iint \Big\{ k\Big(\exp_z(s_\tau \Phi^\tau(z)), \exp_w(s_\tau \Phi^\tau(w))\Big) - k(z,w)\Big\}\rho^\tau(\mathrm{d}z)\rho^\tau(\mathrm{d}w) \\ &\quad - \iint \Big\{ k\Big(\exp_z(s_\tau \Phi^\tau(z)), w\Big) - k(z,w)\Big\}\rho^\tau(\mathrm{d}z)\varrho(\mathrm{d}w).\end{aligned}$$

Moreover, we have

$$\begin{aligned}&\int \langle \nabla_d[\mathbf{m}_{\rho^\tau} - \mathbf{m}_\varrho](z), \Phi^\tau(z)\rangle_z \rho^\tau(\mathrm{d}z) \\ &= \iint \langle \nabla_d^1 k(z,w), \Phi^\tau(z)\rangle_z \rho^\tau(\mathrm{d}z)\rho^\tau(\mathrm{d}w) - \iint \langle \nabla_d^1 k(z,w), \Phi^\tau(z)\rangle_z \rho^\tau(\mathrm{d}z)\varrho(\mathrm{d}w) \\ &= \frac{1}{2}\left[\iint \langle \nabla_d^1 k(z,w), \Phi^\tau(z)\rangle_z \rho^\tau(\mathrm{d}z)\rho^\tau(\mathrm{d}w) + \iint \langle \nabla_d^2 k(z,w), \Phi^\tau(w)\rangle_w \rho^\tau(\mathrm{d}z)\rho^\tau(\mathrm{d}w)\right] \\ &\quad - \iint \langle \nabla_d^1 k(z,w), \Phi^\tau(z)\rangle_z \rho^\tau(\mathrm{d}z)\varrho(\mathrm{d}w),\end{aligned}$$

where the last equality is due to the symmetry of k and relation (3.6). Here $\nabla_d^1 k(z,w)$ and $\nabla_d^2 k(z,w)$ respectively denote the gradient ∇_d of the function $z \mapsto k(z,w)$ and $w \mapsto k(z,w)$. Therefore, it follows that

$$\begin{aligned}&\mathcal{F}[\rho^{\tau+1}] - \mathcal{F}[\rho^\tau] - s_\tau \int \langle \nabla_d[\mathbf{m}_{\rho^\tau} - \mathbf{m}_\varrho](z), \Phi^\tau(z)\rangle_z \rho^\tau(\mathrm{d}z) \\ &\quad = \frac{1}{2}\iint \Big\{ k\Big(\exp_z(s_\tau \Phi^\tau(z)), \exp_w(s_\tau \Phi^\tau(w))\Big) - k(z,w) \\ &\qquad\qquad - \Big[\langle \nabla_d^1 k(z,w), s_\tau \Phi^\tau(z)\rangle_z + \langle \nabla_d^2 k(z,w), s_\tau \Phi^\tau(w)\rangle_w\Big]\Big\}\,\rho^\tau(\mathrm{d}z)\rho^\tau(\mathrm{d}w) \\ &\quad\quad - \iint \Big\{ k\Big(\exp_z(s_\tau \Phi^\tau(z)), w\Big) - k(z,w) - \langle \nabla_d^1 k(z,w), s_\tau \Phi^\tau(z)\rangle_z\Big\}\rho^\tau(\mathrm{d}z)\varrho(\mathrm{d}w).\end{aligned}$$

As $s_\tau \in (0, \varepsilon_0]$, we can now use the assumption that k is a Lipschitz-gradient kernel with constant L to obtain

$$\begin{aligned}&\mathcal{F}[\rho^{\tau+1}] - \mathcal{F}[\rho^\tau] + s_\tau \int_{\mathcal{Z}} \|\Phi^\tau(z)\|_z^2 \rho^\tau(\mathrm{d}z) \\ &\le \frac{L}{2}\iint \Big[\|s_\tau \Phi^\tau(z)\|_z^2 + \|s_\tau \Phi^\tau(w)\|_w^2\Big]\rho^\tau(\mathrm{d}z)\rho^\tau(\mathrm{d}w) + L\iint \|s_\tau \Phi^\tau(z)\|_z^2 \rho^\tau(\mathrm{d}z)\varrho(\mathrm{d}w) \\ &= 2Ls_\tau^2 \int \|\Phi^\tau(z)\|_z^2 \rho^\tau(\mathrm{d}z).\end{aligned}$$

This gives

$$\mathcal{F}[\rho^{\tau+1}] - \mathcal{F}[\rho^\tau] \le \big(-s_\tau + 2Ls_\tau^2\big)\int_{\mathcal{Z}} \|\Phi^\tau(z)\|_z^2 \rho^\tau(\mathrm{d}z),$$

and the conclusion of the proposition follows. □

Convergence guarantees. For each distribution ρ on $\mathcal{Z}$, let $\mathbb{K}_\rho : \mathcal{H} \to \mathcal{H}$ be the linear operator defined by $\mathbb{K}_\rho f(w_1) := \langle \tilde{\mathbb{K}}_\rho(w_1, \cdot), f(\cdot)\rangle_{\mathcal{H}}$ with $\tilde{\mathbb{K}}_\rho : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ being given by

$$\tilde{\mathbb{K}}_\rho(w_1, w_2) = \int_{\mathcal{Z}} \langle \nabla_d^1 k(z, w_1), \nabla_d^1 k(z, w_2)\rangle_z \,\rho(\mathrm{d}z) \quad \text{for} \quad w_1,\, w_2 \in \mathcal{Z}.$$

The next result gives some basic properties of the operator $\mathbb{K}_\rho$.

Lemma A.7. *For a differentiable kernel k and for $\rho \in \mathcal{P}(\mathcal{Z})$, we have*

i) $\mathbb{K}_\rho f(w) = \int_{\mathcal{Z}} \langle \nabla_d^1 k(z,w), \nabla_d f(z)\rangle_z \,\rho(\mathrm{d}z)$ *for* $f \in \mathcal{H}$.

ii) $\langle \mathbb{K}_\rho f, g\rangle_{\mathcal{H}} = \int_{\mathcal{Z}} \langle \nabla_d f, \nabla_d g\rangle_z \, \rho(\mathrm{d}z)$ *for every* $f,\, g \in \mathcal{H}$*. Consequently, the operator* $\mathbb{K}_\rho$ *is symmetric and positive, and hence its spectrum is contained in* $[0, +\infty)$*.*

Proof of Lemma A.7. By using the definition of the Riemannian metric $\langle \cdot, \cdot\rangle_z$ given in (3.3), it can be verified for $f \in \mathcal{H}$ that

$$\Big\langle \langle \nabla_d^1 k(z, w), \nabla_d^1 k(z, \cdot)\rangle_z, f(\cdot)\Big\rangle_{\mathcal{H}} = \Big\langle \nabla_d^1 k(z, w), \langle \nabla_d^1 k(z, \cdot), f(\cdot)\rangle_{\mathcal{H}}\Big\rangle_z.$$

As $f(z) = \langle k(z, \cdot), f(\cdot)\rangle_{\mathcal{H}}$, we moreover have $\nabla_d f(z) = \langle \nabla_d^1 k(z, \cdot), f(\cdot)\rangle_{\mathcal{H}}$. Therefore,

$$\Big\langle \langle \nabla_d^1 k(z, w), \nabla_d^1 k(z, \cdot)\rangle_z, f(\cdot)\Big\rangle_{\mathcal{H}} = \big\langle \nabla_d^1 k(z, w), \nabla_d f(z)\big\rangle_z. \tag{A.9}$$

Using the definition of $\mathbb{K}_\rho$ and (A.9), we obtain

$$\begin{aligned}\mathbb{K}_\rho f(w) &= \Big\langle \int \langle \nabla_d^1 k(z, w), \nabla_d^1 k(z, \cdot)\rangle_z \, \rho(\mathrm{d}z), f(\cdot)\Big\rangle_{\mathcal{H}} \\ &= \int \Big\langle \langle \nabla_d^1 k(z, w), \nabla_d^1 k(z, \cdot)\rangle_z, f(\cdot)\Big\rangle_{\mathcal{H}} \rho(\mathrm{d}z) \\ &= \int \big\langle \nabla_d^1 k(z, w), \nabla_d f(z)\big\rangle_z \, \rho(\mathrm{d}z),\end{aligned}$$

which gives i). Now for $f,\, g \in \mathcal{H}$, we can use part i) and similar arguments leading to (A.9) to obtain

$$\begin{aligned}\langle \mathbb{K}_\rho f, g\rangle_{\mathcal{H}} &= \Big\langle \int \big\langle \nabla_d^1 k(z, \cdot), \nabla_d f(z)\big\rangle_z \, \rho(\mathrm{d}z), g(\cdot)\Big\rangle_{\mathcal{H}} \\ &= \int \Big\langle \langle \nabla_d^1 k(z, \cdot), \nabla_d f(z)\rangle_z, g(\cdot)\Big\rangle_{\mathcal{H}} \rho(\mathrm{d}z) \\ &= \int \Big\langle \langle \nabla_d^1 k(z, \cdot), g(\cdot)\rangle_{\mathcal{H}}, \nabla_d f(z)\Big\rangle_z \rho(\mathrm{d}z) = \int \big\langle \nabla_d g(z), \nabla_d f(z)\big\rangle_z \, \rho(\mathrm{d}z).\end{aligned}$$

This implies in particular that the operator $\mathbb{K}_\rho$ is symmetric (i.e. $\langle \mathbb{K}_\rho f, g\rangle_{\mathcal{H}} = \langle \mathbb{K}_\rho g, f\rangle_{\mathcal{H}}$ for $f,\, g \in \mathcal{H}$) and positive (i.e. $\langle \mathbb{K}_\rho f, f\rangle_{\mathcal{H}} \geq 0$ for $f \in \mathcal{H}$). Since any symmetric, positive, and linear operator must have nonnegative eigenvalues, we have completed the proof. □

Our next result gives a quantified decay rate for the objective function.

Proposition A.8 (Objective value decay)**.** *There hold:*

i) Let ρ_t *be given by* (4.1)*, and let* $\lambda_t \geq 0$ *be any constant satisfying*

$$\langle \mathbb{K}_{\rho_t} f_t, f_t\rangle_{\mathcal{H}} \geq \lambda_t \|f_t\|_{\mathcal{H}}^2 \quad \textit{with} \quad f_t := \mathbf{m}_{\rho_t} - \mathbf{m}_{\varrho}. \tag{A.10}$$

Then $\mathcal{F}[\rho_t] \leq \mathcal{F}[\rho_0] \exp\big(-2\int_0^t \lambda_s \mathrm{d}s\big)$ *for any* $t \geq 0$*. In particular,* $\lim_{t\to\infty} \mathrm{MMD}(\rho_t, \varrho) = 0$ *if* $\int_0^\infty \lambda_t \, \mathrm{d}t = +\infty$*.*

ii) Let ρ^τ *be given by scheme* (4.2)*, and* $\lambda_\tau \geq 0$ *be any constant satisfying*

$$\langle \mathbb{K}_{\rho_t} f^\tau, f^\tau\rangle_{\mathcal{H}} \geq \lambda_\tau \|f^\tau|_{\mathcal{H}}^2 \quad \textit{with} \quad f^\tau := \mathbf{m}_{\rho^\tau} - \mathbf{m}_{\varrho}.$$

Assume that k *is a Lipschitz-gradient kernel and step size* s_τ *satisfies* $s_\tau \lambda_\tau < 1$*, then we have* $\mathcal{F}[\rho^{\tau+1}] \leq \mathcal{F}[\rho^0] \exp\big(-\sum_{i=0}^{\tau} s_i \lambda_i\big)$ *for any* $\tau \geq 0$*. In particular,* $\lim_{\tau\to\infty} \mathrm{MMD}(\rho^\tau, \varrho) = 0$ *if* $\sum_{\tau=0}^\infty s_\tau \lambda_\tau = +\infty$*.*

Condition $\int_0^\infty \lambda_t \, \mathrm{d}t = +\infty$ guaranteeing the convergence in MMD holds true for example if $\lambda_t \geq c\, t^{-1}$ for some constant $c > 0$ and for large t. We note also that Condition (A.10) is satisfied if λ_t is chosen to be the minimum eigenvalue of operator $\mathbb{K}_{\rho_t}$. Thus Proposition A.8 implies in particular that ρ_t globally converges in MMD if the minimum eigenvalue λ_t of operator $\mathbb{K}_{\rho_t}$ satisfies the integrability condition $\int_0^\infty \lambda_t \, \mathrm{d}t = +\infty$. The proof of Proposition A.8 relies on the following proposition, which shows that the dynamic of the mean embedding is governed by the equation $\partial_t(\mathbf{m}_{\rho_t} - \mathbf{m}_{\varrho}) = -\mathbb{K}_{\rho_t}(\mathbf{m}_{\rho_t} - \mathbf{m}_{\varrho})$.

Proposition A.9 (Dynamic of the mean embedding)**.** *Let $t \in [0,\infty) \longmapsto \rho_t$ be the gradient flow given by equation* (4.1)*. For each $t \geq 0$, take $f_t := \mathbf{m}_{\rho_t} - \mathbf{m}_{\varrho}$. Then f_t is a solution of the linear partial differential equation*

$$\partial_t f_t = -\mathbb{K}_{\rho_t} f_t \quad \textit{in} \quad [0,\infty)\times\mathcal{Z}. \tag{A.11}$$

Proof of Proposition A.9. From the definition of the mean embedding and by using equation (4.1), we have

$$\begin{aligned}\partial_t f_t(w) = \partial_t \mathbf{m}_{\rho_t}(w) = \partial_t \int_{\mathcal{Z}} k(z,w)\,\rho_t(\mathrm{d}z) &= \int_{\mathcal{Z}} k(z,w)\,\partial_t\rho_t(\mathrm{d}z)\\ &= \int_{\mathcal{Z}} k(z,w)\,\mathrm{div}_d(\rho_t\nabla_d f_t)(\mathrm{d}z).\end{aligned}$$

Using the definition of the divergence operator div_d at the end of Section 3.1, we further obtain

$$\partial_t f_t(w) = -\int_{\mathcal{Z}} \langle \nabla_d^1 k(z,w), \nabla_d f_t(z)\rangle_z\,\rho_t(\mathrm{d}z).$$

It then follows from part i) of Lemma A.7 that $\partial_t f_t(w) = -\mathbb{K}_{\rho_t} f_t(w)$. This completes the proof. □

We are now ready to present the proof of Proposition A.8.

Proof of Proposition A.8. Let $f_t := \mathbf{m}_{\rho_t} - \mathbf{m}_{\varrho}$. Then we have from Proposition 4.2 and part ii) of Lemma A.7 that $\partial_t \|f_t\|_{\mathcal{H}}^2 = -2\langle \mathbb{K}_{\rho_t} f_t, f_t\rangle_{\mathcal{H}}$. But as

$$\langle \mathbb{K}_{\rho_t} f_t, f_t\rangle_{\mathcal{H}} \geq \lambda_t \|f_t\|_{\mathcal{H}}^2$$

by Condition A.10, we infer that $\partial_t \|f_t\|_{\mathcal{H}}^2 \leq -2\lambda_t \|f_t\|_{\mathcal{H}}^2$, and hence $\partial_t\Big(\log\|f_t\|_{\mathcal{H}}^2\Big) \leq -2\lambda_t$. By integrating from 0 to t, one gets $\log\|f_t\|_{\mathcal{H}}^2 - \log\|f_0\|_{\mathcal{H}}^2 \leq -2\int_0^t \lambda_s\,\mathrm{d}s$. We next take exponential to obtain

$$\|f_t\|_{\mathcal{H}}^2 \leq \|f_0\|_{\mathcal{H}}^2 \exp\Big(-2\int_0^t \lambda_s\,\mathrm{d}s\Big).$$

This can be rewritten as $\mathcal{F}[\rho_t] \leq \mathcal{F}[\rho_0]\exp\big(-2\int_0^t \lambda_s \mathrm{d}s\big)$ for $t \geq 0$. In particular, $\mathcal{F}[\rho_t]$ (and hence $\mathrm{MMD}(\rho_t,\varrho)$) tends to zero if $\int_0^\infty \lambda_t\,\mathrm{d}t = +\infty$. This completes the proof for part i).

To prove ii), let $f^\tau := \mathbf{m}_{\rho^\tau} - \mathbf{m}_{\varrho}$. Notice that in contrast to the continuous case, upper indices are used for f^τ and ρ^τ in the discrete case. Then by using Proposition A.6 together with part ii) of Lemma A.7 and the assumption $s_\tau \in (0, \frac{1}{4L}]$ we have

$$\mathcal{F}[\rho^{\tau+1}] - \mathcal{F}[\rho^\tau] \leq -\frac{1}{2}s_\tau\langle \mathbb{K}_{\rho^\tau} f^\tau, f^\tau\rangle_{\mathcal{H}}.$$

But as $\langle \mathbb{K}_{\rho^\tau} f^\tau, f^\tau\rangle_{\mathcal{H}} \geq \lambda_\tau\|f^\tau\|_{\mathcal{H}}^2$ due to our assumption, we obtain $\mathcal{F}[\rho^{\tau+1}] - \mathcal{F}[\rho^\tau] \leq -s_\tau\lambda_\tau\mathcal{F}[\rho^\tau]$, or

$$\mathcal{F}[\rho^{\tau+1}] \leq (1 - s_\tau\lambda_\tau)\mathcal{F}[\rho^\tau]$$

for every $\tau \geq 0$. As $1 - s_\tau\lambda_\tau > 0$, it follows by iteration that $\mathcal{F}[\rho^{\tau+1}] \leq \mathcal{F}[\rho^0]\prod_{i=0}^{\tau}(1 - s_i\lambda_i)$. Due to $1 - x \leq \exp(-x)$ for every $x \geq 0$, we infer that $\mathcal{F}[\rho^{\tau+1}] \leq \mathcal{F}[\rho^0]\exp\big(-\sum_{i=0}^{\tau} s_i\lambda_i\big)$ for $\tau \geq 0$. In particular, $\mathcal{F}[\rho^\tau]$ (and hence $\mathrm{MMD}(\rho^\tau,\varrho)$) tends to zero if $\sum_{\tau=0}^{\infty} s_\tau\lambda_\tau = +\infty$. □

For Proposition 4.4

Proof of Proposition 4.4. Let $h(z) := \exp_z(s_\tau \Phi^\tau(z))$ for $z \in Z$. Then $\rho^{\tau+1}$ can be expressed as

$$\rho^{\tau+1} = h_\# \rho^{\tau,\beta_\tau} = (h \circ f^{\beta_\tau})_\# (\rho^\tau \otimes g).$$

By the computation at the beginning of the proof of Lemma 4.3 using Lemma A.2, we obtain

$$\begin{aligned}
\mathcal{F}[\rho^{\tau+1}] - \mathcal{F}[\rho^\tau] &= \frac{1}{2}\left[\mathrm{MMD}\Big((h \circ f^{\beta_\tau})_\#(\rho^\tau \otimes g), \varrho\Big)^2 - \mathrm{MMD}(\rho^\tau, \varrho)^2\right] \\
&= \frac{1}{2}\iiiint \Big\{k\Big(h(f^{\beta_\tau}(z,u)), h(f^{\beta_\tau}(w,v))\Big) - k(z,w)\Big\}\rho^\tau(\mathrm{d}z)g(\mathrm{d}u)\rho^\tau(\mathrm{d}w)g(\mathrm{d}v) \\
&\quad - \iiint \Big\{k\Big(h(f^{\beta_\tau}(z,u)), w\Big) - k(z,w)\Big\}\rho^\tau(\mathrm{d}z)g(\mathrm{d}u)\varrho(\mathrm{d}w).
\end{aligned}$$

Moreover, we have

$$\begin{aligned}
I &:= \int \langle \nabla_d[\mathbf{m}_{\rho^\tau} - \mathbf{m}_\varrho](z), \Phi^\tau(z)\rangle_z \rho^{\tau,\beta_\tau}(\mathrm{d}z) \\
&= \iint \langle \nabla_d^1 k(z,w), \Phi^\tau(z)\rangle_z \rho^{\tau,\beta_\tau}(\mathrm{d}z)\rho^\tau(\mathrm{d}w) - \iint \langle \nabla_d^1 k(z,w), \Phi^\tau(z)\rangle_z \rho^{\tau,\beta_\tau}(\mathrm{d}z)\varrho(\mathrm{d}w) \\
&= \frac{1}{2}\iint \langle \nabla_d^1 k(z,w), \Phi^\tau(z)\rangle_z \rho^{\tau,\beta_\tau}(\mathrm{d}z)\rho^\tau(\mathrm{d}w) + \frac{1}{2}\iint \langle \nabla_d^2 k(z,w), \Phi^\tau(w)\rangle_w \rho^{\tau,\beta_\tau}(\mathrm{d}w)\rho^\tau(\mathrm{d}z) \\
&\quad - \iint \langle \nabla_d^1 k(z,w), \Phi^\tau(z)\rangle_z \rho^{\tau,\beta_\tau}(\mathrm{d}z)\varrho(\mathrm{d}w) \\
&= \frac{1}{2}\iiint \Big\langle \nabla_d^1 k(f^{\beta_\tau}(z,u), w), \Phi^\tau(f^{\beta_\tau}(z,u))\Big\rangle_{f^{\beta_\tau}(z,u)} \rho^\tau(\mathrm{d}z)g(\mathrm{d}u)\rho^\tau(\mathrm{d}w) \\
&\quad + \frac{1}{2}\iiint \Big\langle \nabla_d^2 k(z, f^{\beta_\tau}(w,v)), \Phi^\tau(f^{\beta_\tau}(w,v))\Big\rangle_{f^{\beta_\tau}(w,v)} \rho^\tau(\mathrm{d}w)g(\mathrm{d}v)\rho^\tau(\mathrm{d}z) \\
&\quad - \iiint \Big\langle \nabla_d^1 k(f^{\beta_\tau}(z,u), w), \Phi^\tau(f^{\beta_\tau}(z,u))\Big\rangle_{f^{\beta_\tau}(z,u)} \rho^\tau(\mathrm{d}z)g(\mathrm{d}u)\varrho(\mathrm{d}w),
\end{aligned}$$

where the third equality is due to the symmetry of k and relation (3.6). Therefore, it follows that

$$\begin{aligned}
&\mathcal{F}[\rho^{\tau+1}] - \mathcal{F}[\rho^\tau] - s_\tau I \\
&= \frac{1}{2}\iiiint \Big\{k\Big(h(f^{\beta_\tau}(z,u)), h(f^{\beta_\tau}(w,v))\Big) - k(z,w) \\
&\qquad\qquad - \Big[\langle \nabla_d^1 k(f^{\beta_\tau}(z,u), w), s_\tau \Phi^\tau(f^{\beta_\tau}(z,u))\rangle_{f^{\beta_\tau}(z,u)} \\
&\qquad\qquad + \langle \nabla_d^2 k(z, f^{\beta_\tau}(w,v)), s_\tau \Phi^\tau(f^{\beta_\tau}(w,v))\rangle_{f^{\beta_\tau}(w,v)}\Big]\Big\}\,\rho^\tau(\mathrm{d}z)g(\mathrm{d}u)\rho^\tau(\mathrm{d}w)g(\mathrm{d}v) \\
&\quad - \iiint \Big\{k\Big(h(f^{\beta_\tau}(z,u)), w\Big) - k(z,w) - \langle \nabla_d^1 k(f^{\beta_\tau}(z,u), w), s_\tau \Phi^\tau(f^{\beta_\tau}(z,u))\rangle_{f^{\beta_\tau}(z,u)}\Big\} \\
&\qquad\qquad \rho^\tau(\mathrm{d}z)g(\mathrm{d}u)\varrho(\mathrm{d}w).
\end{aligned}$$

As $h(z) = \exp_z(s_\tau \Phi^\tau(z))$ and $s_\tau \in (0, \varepsilon_0]$, we can now use the Lipschitz-gradient condition (A.8) for k to obtain

$$\begin{aligned}
&\mathcal{F}[\rho^{\tau+1}] - \mathcal{F}[\rho^\tau] - s_\tau I \\
&\le \frac{L}{2}\iiiint \Big[\|s_\tau \Phi^\tau(f^{\beta_\tau}(z,u))\|^2_{f^{\beta_\tau}(z,u)} + \|s_\tau \Phi^\tau(f^{\beta_\tau}(w,v))\|^2_{f^{\beta_\tau}(w,v)}\Big]\rho^\tau(\mathrm{d}z)g(\mathrm{d}u)\rho^\tau(\mathrm{d}w)g(\mathrm{d}v) \\
&\quad + L\iiint \|s_\tau \Phi^\tau(f^{\beta_\tau}(z,u))\|^2_{f^{\beta_\tau}(z,u)}\rho^\tau(\mathrm{d}z)\varrho(\mathrm{d}w)g(\mathrm{d}u) \\
&= 2Ls_\tau^2 \iint \|\Phi^\tau(f^{\beta_\tau}(z,u))\|^2_{f^{\beta_\tau}(z,u)}\rho^\tau(\mathrm{d}z)g(\mathrm{d}u).
\end{aligned}$$

Using the definition $\rho^{\tau,\beta_\tau} = f_{\#}^{\beta_\tau}(\rho^\tau \otimes g)$ and the fact $I = -\int_{\mathcal{Z}} \|\Phi^\tau(z)\|_z^2 \, \rho^{\tau,\beta_\tau}(\mathrm{d}z)$, we can rewrite this more compactly as

$$\begin{aligned}\mathcal{F}[\rho^{\tau+1}] - \mathcal{F}[\rho^\tau] &\leq -s_\tau\big(1 - 2Ls_\tau\big) \int_{\mathcal{Z}} \|\Phi^\tau(z)\|_z^2 \, \rho^{\tau,\beta_\tau}(\mathrm{d}z) \\ &= -s_\tau\big(1 - 2Ls_\tau\big) \int_{\mathcal{Z}} \|\nabla_d[\mathbf{m}_{\rho^\tau} - \mathbf{m}_\varrho](z)\|_z^2 \, \rho^{\tau,\beta_\tau}(\mathrm{d}z).\end{aligned}$$

This together with condition (4.4) gives

$$\mathcal{F}[\rho^{\tau+1}] \leq (1 - a_\tau)\mathcal{F}[\rho^\tau] \quad \text{with} \quad a_\tau \coloneqq \lambda s_\tau\big(1 - 2Ls_\tau\big)\beta_\tau^2.$$

In particular, we must have $a_i \leq 1$. By iterating this estimate, we obtain

$$\mathcal{F}[\rho^{\tau+1}] \leq \mathcal{F}[\rho^0] \prod_{i=0}^{\tau} (1 - a_i). \tag{A.12}$$

Due to $1 - x \leq \exp(-x)$ for every number $x \geq 0$, we get $\prod_{i=0}^{\tau}(1 - a_i) \leq \exp(-\sum_{i=0}^{\tau} a_i)$. This together with (A.12) yields the conclusion of the proposition. □

B Implementation and Experiment Details

We use $\nabla_{\mathbb{B}}^1 k(x, \mu, \Sigma, w)$ to denote the last component in (3.6) for the gradient ∇_d of the function $(x, \mu, \Sigma) \mapsto k(x, \mu, \Sigma, w)$. Precisely,

$$\nabla_{\mathbb{B}}^1 k(x, \mu, \Sigma, w) \coloneqq 2[\nabla_\Sigma k(x, \mu, \Sigma, w)]\Sigma + 2\Sigma[\nabla_\Sigma k(x, \mu, \Sigma, w)].$$

B.1 Algorithms

Algorithm 2 Discretized Gradient Flow Algorithm for Scheme (4.2) – Detailed Version of Algorithm 1

```
Input: a source distribution $\rho^0 = \frac{1}{N}\sum_{i=1}^N \delta_{(x_i^0,\mu_i^0,\Sigma_i^0)}$, a sample $\frac{1}{M}\sum_{j=1}^M \delta_{(\bar{x}_j,\bar{\mu}_j,\bar{\Sigma}_j)}$ for the target distribution $\varrho$, a number $T$ of iterations for training, a sequence of step sizes $s_\tau > 0$ with $\tau = 0, 1, ..., T$, and a kernel $k$.
Initialization:
Compute $(\bar{\Psi}_1, \bar{\Psi}_2, \bar{\Psi}_3)(x, \mu, \Sigma) = \frac{1}{M}\sum_{j=1}^M (\nabla_x, \nabla_\mu, \nabla_{\mathbb{B}}^1) k(x, \mu, \Sigma, \bar{x}_j, \bar{\mu}_j, \bar{\Sigma}_j)$
$\tau \leftarrow 0$
while $\tau < T$ do
  Compute $(\Psi_1^\tau, \Psi_2^\tau, \Psi_3^\tau)(x, \mu, \Sigma) = \frac{1}{N}\sum_{i=1}^N (\nabla_x, \nabla_\mu, \nabla_{\mathbb{B}}^1) k(x, \mu, \Sigma, x_i^\tau, \mu_i^\tau, \Sigma_i^\tau)$
  for $i = 1, \ldots, N$ do
    $x_i^{\tau+1} \leftarrow x_i^\tau + s_\tau(\bar{\Psi}_1 - \Psi_1^\tau)(x_i^\tau, \mu_i^\tau, \Sigma_i^\tau)$
    $\mu_i^{\tau+1} \leftarrow \mu_i^\tau + s_\tau(\bar{\Psi}_2 - \Psi_2^\tau)(x_i^\tau, \mu_i^\tau, \Sigma_i^\tau)$
    $\Sigma_i^{\tau+1} \leftarrow \Big(I + s_\tau \mathrm{L}_{\Sigma_i^\tau}\big[(\bar{\Psi}_3 - \Psi_3^\tau)(x_i^\tau, \mu_i^\tau, \Sigma_i^\tau)\big]\Big)\Sigma_i^\tau\Big(I + s_\tau \mathrm{L}_{\Sigma_i^\tau}[(\bar{\Psi}_3 - \Psi_3^\tau)(x_i^\tau, \mu_i^\tau, \Sigma_i^\tau)]\Big)$
  end for
  Set $\tau \leftarrow \tau + 1$
end while
Output: $\rho^T = \frac{1}{N}\sum_{i=1}^N \delta_{(x_i^T,\mu_i^T,\Sigma_i^T)}$
```

Algorithm 3 Discretized Gradient Flow Algorithm for Scheme (4.3)

```
Input: a source distribution $\rho^0 = \frac{1}{N}\sum_{i=1}^{N}\delta_{(x_i,\mu_i,\Sigma_i)}$, a target distribution $\varrho = \frac{1}{M}\sum_{j=1}^{M}\delta_{(\bar{x}_j,\bar{\mu}_j,\bar{\Sigma}_j)}$, number of iterations $T$, step sizes $s_\tau > 0$, noise levels $\beta_\tau$, and a kernel $k$.
Initialization:
Compute $(\bar{\Psi}_1, \bar{\Psi}_2, \bar{\Psi}_3)(x, \mu, \Sigma) = \frac{1}{M}\sum_{j=1}^{M}(\nabla_x, \nabla_\mu, \nabla_{\mathbb{B}}^1)k(x, \mu, \Sigma, \bar{x}_j, \bar{\mu}_j, \bar{\Sigma}_j)$
$\tau \leftarrow 0$
while $\tau < T$ do
  Compute $(\Psi_1^\tau, \Psi_2^\tau, \Psi_3^\tau)(x, \mu, \Sigma) = \frac{1}{N}\sum_{j=1}^{N}(\nabla_x, \nabla_\mu, \nabla_{\mathbb{B}}^1)k(x, \mu, \Sigma, x_j^\tau, \mu_j^\tau, \Sigma_j^\tau)$
  for $i = 1, \ldots, N$ do
    Perturb $x_i^{\tau,p} \leftarrow x_i^\tau + \beta_\tau \mathcal{N}_{\mathbb{R}^m}(0,1)$ and $\mu_i^{\tau,p} \leftarrow \mu_i^\tau + \beta_\tau \mathcal{N}_{\mathbb{R}^n}(0,1)$
    Set $S \leftarrow \beta_\tau \mathcal{N}_{\mathbb{S}^n}(0,1)$ and perturb $\Sigma_i^{\tau,p} \leftarrow (I + \mathrm{L}_{\Sigma_i^\tau}[S])\Sigma_i^\tau(I + \mathrm{L}_{\Sigma_i^\tau}[S])$
    $x_i^{\tau+1} \leftarrow x_i^{\tau,p} + s_\tau(\bar{\Psi}_1 - \Psi_1^\tau)(x_i^{\tau,p}, \mu_i^{\tau,p}, \Sigma_i^{\tau,p})$
    $\mu_i^{\tau+1} \leftarrow \mu_i^{\tau,p} + s_\tau(\bar{\Psi}_2 - \Psi_2^\tau)(x_i^{\tau,p}, \mu_i^{\tau,p}, \Sigma_i^{\tau,p})$
    $\Sigma_i^{\tau+1} \leftarrow \Big(I + s_\tau \mathrm{L}_{\Sigma_i^{\tau,p}}\big[(\bar{\Psi}_3 - \Psi_3^\tau)(x_i^{\tau,p}, \mu_i^{\tau,p}, \Sigma_i^{\tau,p})\big]\Big)\Sigma_i^{\tau,p}\Big(I + s_\tau \mathrm{L}_{\Sigma_i^{\tau,p}}[(\bar{\Psi}_3 - \Psi_3^\tau)(x_i^{\tau,p}, \mu_i^{\tau,p}, \Sigma_i^{\tau,p})]\Big)$
  end for
  Set $\tau \leftarrow \tau + 1$
end while
Output: $\rho^T = \frac{1}{N}\sum_{i=1}^{N}\delta_{(x_i^T,\mu_i^T,\Sigma_i^T)}$
```

B.2 Kernel and Its Gradient for Implementation

We use the kernel k given by:

$$k\left((x,\mu,\Sigma),(\bar{x},\bar{\mu},\bar{\Sigma})\right) := \exp\left(-\alpha\|x-\bar{x}\|_2^2\right)\exp\left(-\beta\|\mu-\bar{\mu}\|_2^2\right)\exp\left(-\gamma\|\Sigma-\bar{\Sigma}\|_2^2\right),$$

where α, β and γ are parameters (bandwidth) of the kernel. We note that this kernel is characteristic by (Szabó & Sriperumbudur, 2018, Theorem 4). Then its standard Euclidean gradient is given by

$$\nabla_{(x,\mu,\Sigma)}k\left((x,\mu,\Sigma),(\bar{x},\bar{\mu},\bar{\Sigma})\right) = -2\exp\left(-\alpha\|x-\bar{x}\|_2^2 - \beta\|\mu-\bar{\mu}\|_2^2 - \gamma\|\Sigma-\bar{\Sigma}\|_2^2\right)\begin{bmatrix}\alpha(x-\bar{x})\\ \beta(\mu-\bar{\mu})\\ \gamma(\Sigma-\bar{\Sigma})\end{bmatrix}.$$

Thus by plugging into formula (3.6), we obtain

$$\begin{aligned}&\nabla_d^1 k\left((x,\mu,\Sigma),(\bar{x},\bar{\mu},\bar{\Sigma})\right)\\ &\qquad = -2\exp\left(-\alpha\|x-\bar{x}\|_2^2 - \beta\|\mu-\bar{\mu}\|_2^2 - \gamma\|\Sigma-\bar{\Sigma}\|_2^2\right)\begin{bmatrix}\alpha(x-\bar{x})\\ \beta(\mu-\bar{\mu})\\ 2\gamma(2\Sigma^2 - \Sigma\bar{\Sigma} - \bar{\Sigma}\Sigma)\end{bmatrix}.\end{aligned}$$

That is,

$$\begin{aligned}&\nabla_x k\left((x,\mu,\Sigma),(\bar{x},\bar{\mu},\bar{\Sigma})\right) = -2\exp\left(-\alpha\|x-\bar{x}\|_2^2 - \beta\|\mu-\bar{\mu}\|_2^2 - \gamma\|\Sigma-\bar{\Sigma}\|_2^2\right)\alpha(x-\bar{x}),\\ &\nabla_\mu k\left((x,\mu,\Sigma),(\bar{x},\bar{\mu},\bar{\Sigma})\right) = -2\exp\left(-\alpha\|x-\bar{x}\|_2^2 - \beta\|\mu-\bar{\mu}\|_2^2 - \gamma\|\Sigma-\bar{\Sigma}\|_2^2\right)\beta(\mu-\bar{\mu}),\\ &\nabla_{\mathbb{B}}^1 k\left((x,\mu,\Sigma),(\bar{x},\bar{\mu},\bar{\Sigma})\right)\\ &\quad = -2\exp\left(-\alpha\|x-\bar{x}\|_2^2 - \beta\|\mu-\bar{\mu}\|_2^2 - \gamma\|\Sigma-\bar{\Sigma}\|_2^2\right)2\gamma(2\Sigma^2 - \Sigma\bar{\Sigma} - \bar{\Sigma}\Sigma).\end{aligned}$$

B.3 Label Projection

We here propose an approach to recover new samples in the feature-label space from an empirical distribution in the feature-Gaussian space. Consider that after T iterations of the gradient algorithms, we arrive at a distribution $\rho^T = \frac{1}{N}\sum_{i=1}^N \delta_{(x_i^T, \mu_i^T, \Sigma_i^T)}$. We would like to recover a distribution $\nu^T \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ which is induced by ρ^τ. As such, we would like to find a distribution ν^T of the form

$$\nu^T = \frac{1}{N}\sum_{i=1}^N \delta_{(x_i^T, y_i^T)}$$

which corresponds to new target samples $(x_i^T, y_i^T)_{i=1}^N$. Moreover, we are interested in recovering labels within the target domain. To this end, let $\mathcal{Y}_{\text{target}} = \{y \in \mathcal{Y} : \exists j \in [M] \text{ such that } \bar{y}_j = y\}$ be the set of labels in the target dataset, and remind that for any $y \in \mathcal{Y}_{\text{target}}$, $(\bar{\mu}_y, \bar{\Sigma}_y) \in \mathbb{R}^n \times \mathbb{S}_+^n$ is the mean vector and the covariance matrix of the distribution of $\phi(X)$ given $Y = y$. Notice that the mean-covariance embeddings $(\bar{\mu}_y, \bar{\Sigma}_y)$ for $y \in \mathcal{Y}_{\text{target}}$ depend only on the target domain data, and it does not depend on the incumbent distribution ρ^T, nor does it depend on the source dataset. Moreover, we can also compute $\bar{N}_y$ as the number of samples from the target dataset with label y.

Because $(\bar{\mu}_y, \bar{\Sigma}_y)$ is readily computed, we can consider $(\bar{\mu}_y, \bar{\Sigma}_y)$ as the centroids and simply find an assignment that minimizes the sum of distances from (μ_i^T, Σ_i^T) to these centroids. We thus can assign each sample from ρ^T to the the target labels by solving the linear program

$$\begin{aligned} \min \quad & \sum_{i=1}^N \sum_{y \in \mathcal{Y}_{\text{target}}} \theta_{iy} \sqrt{\|\mu_i^T - \bar{\mu}_y\|_2^2 + \mathbb{B}(\Sigma_i^T, \bar{\Sigma}_y)^2} \\ \text{s.t.} \quad & \sum_{y \in \mathcal{Y}_{\text{target}}} \theta_{iy} = \frac{1}{N} \quad \forall i = 1, \ldots, N, \qquad \sum_{i=1}^N \theta_{iy} = \frac{\bar{N}_y}{N} \quad \forall y \in \mathcal{Y}_{\text{target}}, \quad \theta \in [0,1]^{N \times |\mathcal{Y}_{\text{target}}|}, \end{aligned} \tag{B.1}$$

Notice that the assignment problem above does not utilize the information from the covariate x_i^T. Let $\theta^\star$ be the optimal solution of the above optimization problem. Then the dataset $(x_i^T, z_i^T)_{i=1}^N$ recovered from ρ^τ is

$$\nu^T = \frac{1}{N}\sum_{i=1}^N \delta_{(x_i^T, y_i^T)}, \qquad y_i^T = \sum_{y \in \mathcal{Y}_{\text{target}}} y \mathbb{1}(\theta_{iy}^\star = \max\{\theta_i^\star\}) \quad \forall i = 1, \ldots, N.$$

We used the POT library to solve the label recovery problem (B.1).

B.4 Additional Numerical Results

B.4.1 Mixture of Gaussians

We test our algorithm on a toy example: a mixture of Gaussian distributions to another mixture of Gaussian distributions.

The source distribution and target distribution are:

$$\begin{aligned} p_s(x) = & \frac{1}{4}\mathcal{N}\left(\begin{pmatrix} 2.0 \\ -0.3 \end{pmatrix}, \begin{pmatrix} 0.14 & -0.00 \\ -0.00 & 0.22 \end{pmatrix}\right) + \frac{1}{4}\mathcal{N}\left(\begin{pmatrix} 2.0 \\ 0.3 \end{pmatrix}, \begin{pmatrix} 0.43 & 0.18 \\ 0.18 & 0.26 \end{pmatrix}\right) \\ & + \frac{1}{4}\mathcal{N}\left(\begin{pmatrix} -0.3 \\ 2.0 \end{pmatrix}, \begin{pmatrix} 0.66 & 0.02 \\ 0.02 & 0.63 \end{pmatrix}\right) + \frac{1}{4}\mathcal{N}\left(\begin{pmatrix} 0.3 \\ -2.0 \end{pmatrix}, \begin{pmatrix} 0.39 & -0.02 \\ -0.02 & 0.13 \end{pmatrix}\right) \\ p_t(x) = & \frac{1}{4}\mathcal{N}\left(\begin{pmatrix} 2.9 \\ 0.1 \end{pmatrix}, \begin{pmatrix} 0.16 & 0.03 \\ 0.03 & 0.20 \end{pmatrix}\right) + \frac{1}{4}\mathcal{N}\left(\begin{pmatrix} 0.9 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 0.22 & 0.16 \\ 0.16 & 0.46 \end{pmatrix}\right) \\ & + \frac{1}{4}\mathcal{N}\left(\begin{pmatrix} 0.8 \\ 2.2 \end{pmatrix}, \begin{pmatrix} 0.63 & 0.02 \\ 0.02 & 0.66 \end{pmatrix}\right) + \frac{1}{4}\mathcal{N}\left(\begin{pmatrix} 1.4 \\ -1.8 \end{pmatrix}, \begin{pmatrix} 0.18 & 0.10 \\ 0.10 & 0.36 \end{pmatrix}\right) \end{aligned}$$

From each distribution, we sample 25 particles and flow the particles' positions, means, and covariance simultaneously using Alg. 1. After the algorithm converges, we recover the particles' label in the feature-label space by solving problem (B.1).

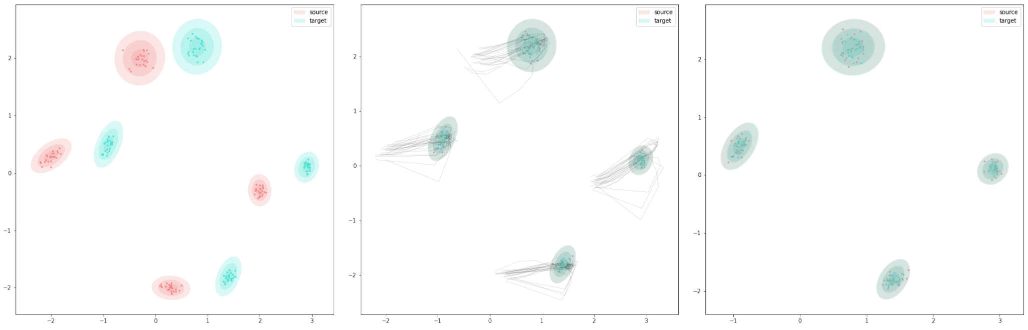

Figure 4: The results of flowing a mixture of 4 Gaussian distributions to a mixture of 4 Gaussian distributions. We demonstrate the initialization (left), the trace of particles in first 200 steps (middle), and the results at step 1000 (right).

We test how our algorithm deals with flowing a mixture of 2 Gaussian distributions to a mixture of 4 Gaussian distributions. From the trace of first 200 steps, we demonstrate that each source Gaussian distribution splits into 2 Gaussian distributions. The source distribution and target distribution are:

$$p_s(x) = \frac{1}{2}\mathcal{N}\left(\begin{pmatrix}0.0\\0.0\end{pmatrix}, \begin{pmatrix}0.18 & -0.24\\-0.24 & 0.70\end{pmatrix}\right) + \frac{1}{2}\mathcal{N}\left(\begin{pmatrix}5.8\\0.0\end{pmatrix}, \begin{pmatrix}0.44 & 0.00\\0.00 & 0.87\end{pmatrix}\right)$$

$$\begin{aligned} p_t(x) = {} & \frac{1}{4}\mathcal{N}\left(\begin{pmatrix}2.0\\0.7\end{pmatrix}, \begin{pmatrix}0.63 & -0.30\\-0.30 & 0.26\end{pmatrix}\right) + \frac{1}{4}\mathcal{N}\left(\begin{pmatrix}2.2\\-0.8\end{pmatrix}, \begin{pmatrix}0.77 & -0.18\\-0.18 & 0.55\end{pmatrix}\right) \\ & + \frac{1}{4}\mathcal{N}\left(\begin{pmatrix}7.0\\0.8\end{pmatrix}, \begin{pmatrix}0.63 & -0.30\\-0.30 & 0.26\end{pmatrix}\right) + \frac{1}{4}\mathcal{N}\left(\begin{pmatrix}7.7\\-0.8\end{pmatrix}, \begin{pmatrix}0.77 & -0.18\\-0.18 & 0.55\end{pmatrix}\right) \end{aligned}$$

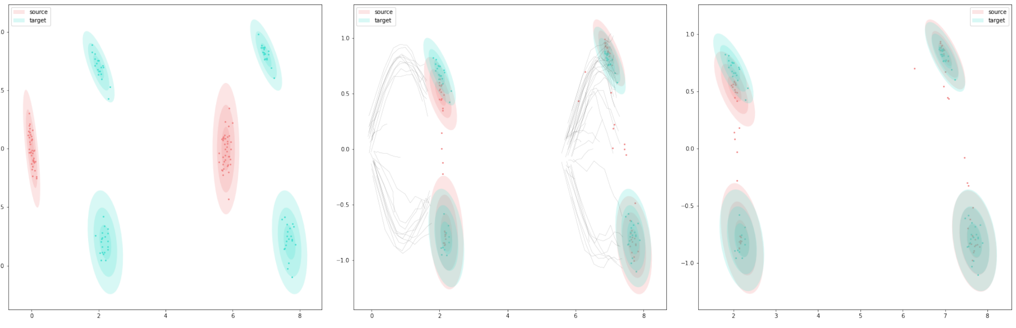

Figure 5: The results of flowing a mixture of 2 Gaussian distributions to a mixture of 4 Gaussian distributions. We demonstrate the initialization (left), the trace of particles in first 200 steps (middle), and the results at step 2400 (right). We use method in Section B.3 to relabel the source data.

B.5 Implementation details

When flowing images in *NIST datasets and flowing a mixture of Gaussians, we use the parameters and methods described in Table 1.

Our method assumes images of each class form one Gaussian distribution. In reality, the data can be a mixture of Gaussian. To satisfy the Gaussianity assumption, in the preprocessing step, we use a clustering method (k-nearest neighbors) and pick only data from one mode for each class. As a consequence, the data used in the experiment satisfies the conditional Gaussian assumption. For example, the images of the digit 1 can have two modes: slanted left or slanted right. In this case, we can generate two labels (1L, 1R), and the methodology developed in this paper can be applied in a straightforward manner. When testing our transfer learning scheme, we apply the same clustering method on the test dataset, so our test set is within the same mode as our training set.

Table 1: Parameters and Optimizer

	*NIST	Gaussian (Figure 4)	Gaussian (Figure 5)
α	0.001	0.3	0.3
β	0.002	0.15	0.1
γ	100	1.0	0.5
initial s_τ	0.3	0.05	0.03
noise level	0.01	0	0.1
T	150	2000	2500
Optimizer	RMSprop Hinton et al. (2012)	RMSprop	RMSprop

We store the preprocessed data and apply dimension reduction method on the data's means and covariance matrices, so the Lyapunov equation is much faster to solve. We use the cluster's mean and covariance matrix to approximate the 1-shot and 5-shor data's mean and covariance matrix. In 1-shot learning, the covariance matrix is an identity matrix. All the code and data are available in the supplementary file.

We use k-nearest neighbors algorithm to solve the labels of the flowed data, as it performs better with the noisy scheme.

B.5.1 Additional results on flows

We conduct additional experiments of flowing between KMNIST and FashionMNIST datasets. The results of our flows are depicted in Fig. 6. In each subfigure, each column represents a snapshot of a certain time-step and the samples flow from the source (left) to the target (right). The number of iterations T that is used to generate the results in Fig. 6 is capped at 140. We also attach the same results as Fig. 2 in high resolution in Fig. 7 and Fig. 8. Each picture illustrates one experiment of gradient flow between two datasets and the samples flow from the source (left) to the target (right).

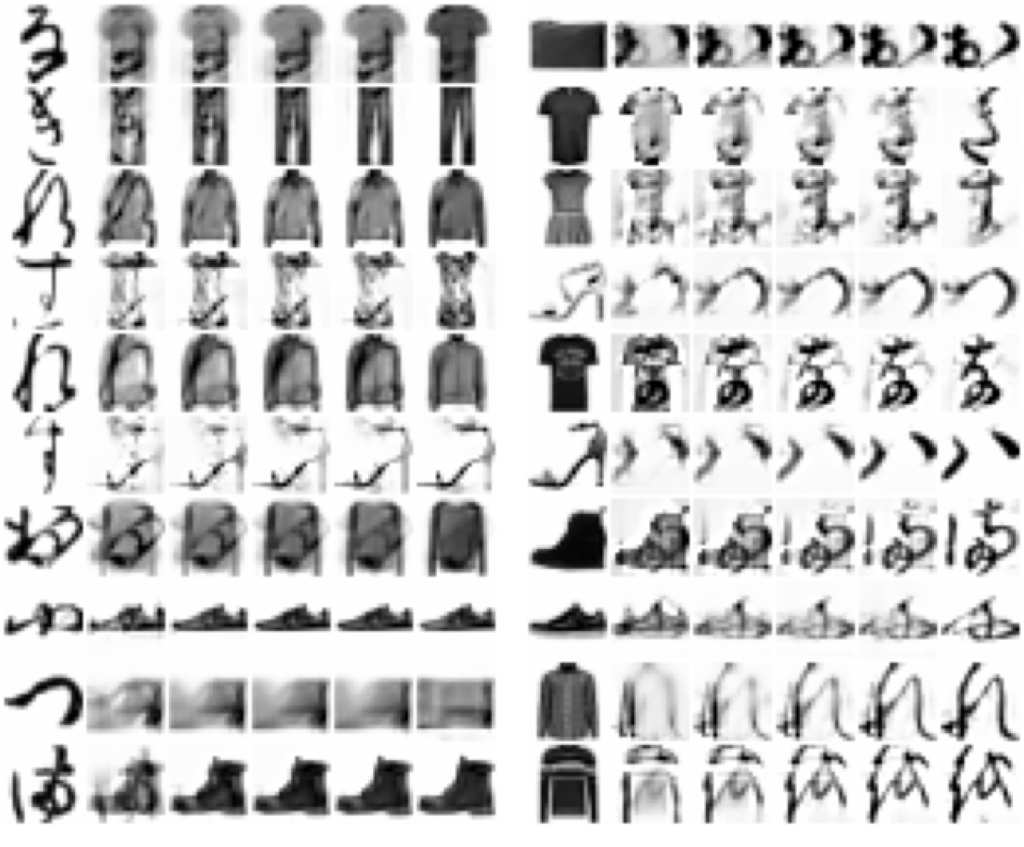

Figure 6: Sample path visualizations between FashionMNIST dataset and KMNIST dataset

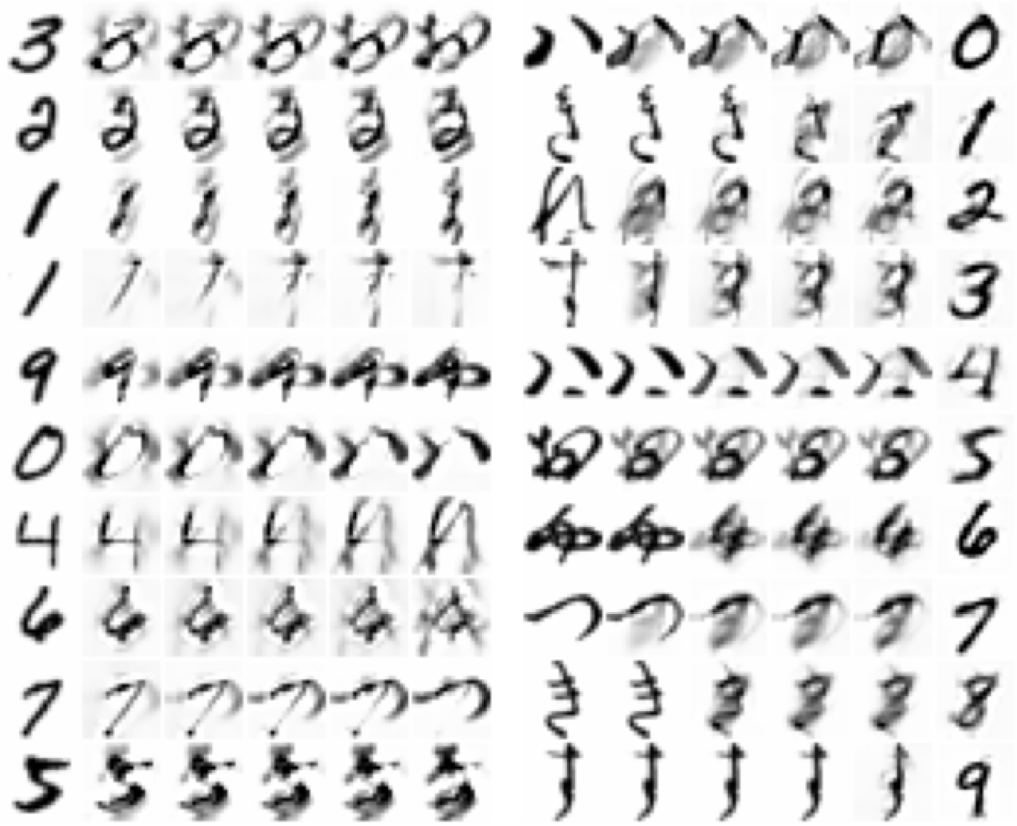

Figure 7: Sample path visualizations between KMNIST dataset and MNIST dataset

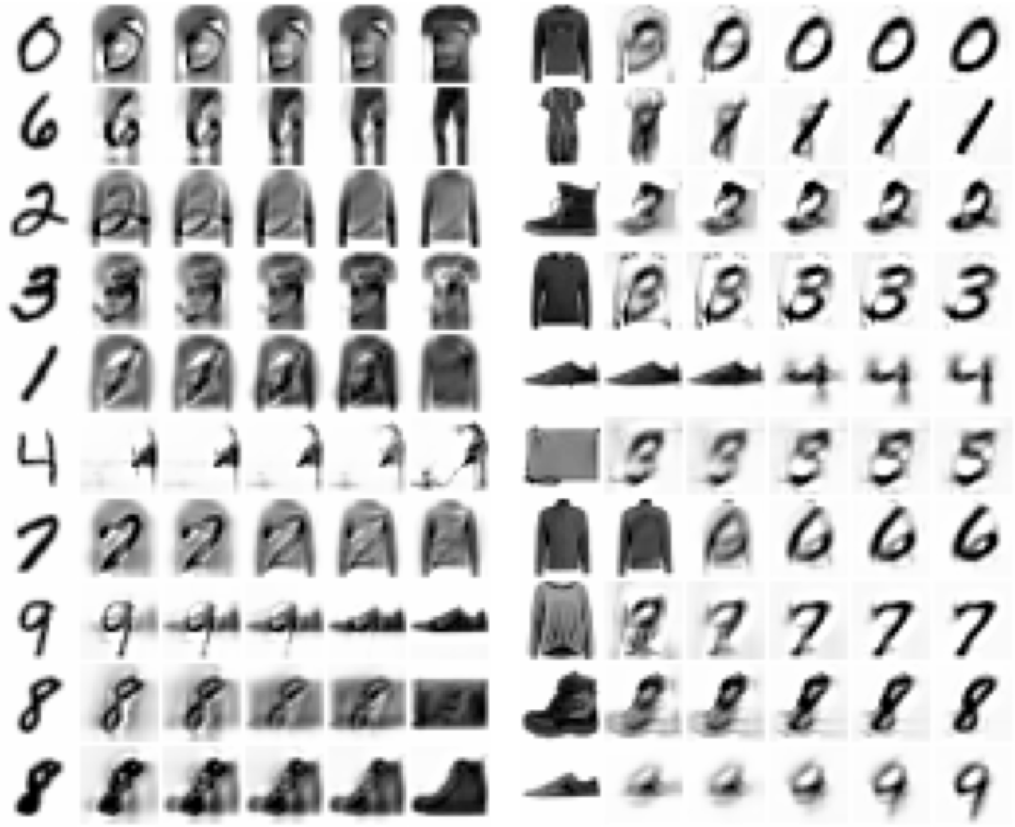

Figure 8: Sample path visualizations between FashionMNIST dataset and MNIST dataset

B.5.2 ADDITIONAL RESULTS ON TRANSFER LEARNING

We transfer a pretrained classifier from FashionMNIST dataset to KMNIST dataset and from KMNIST dataset to FashionMNIST dataset. We use the same model architecture and training settings as in Fig. 3. We illustrate the accuracy and error bars of the 1-shot learning and 5-shot learning in Fig. 9. Our flowed samples increase the accuracy of the transferred classifiers in both 1-shot and 5-shot learning.

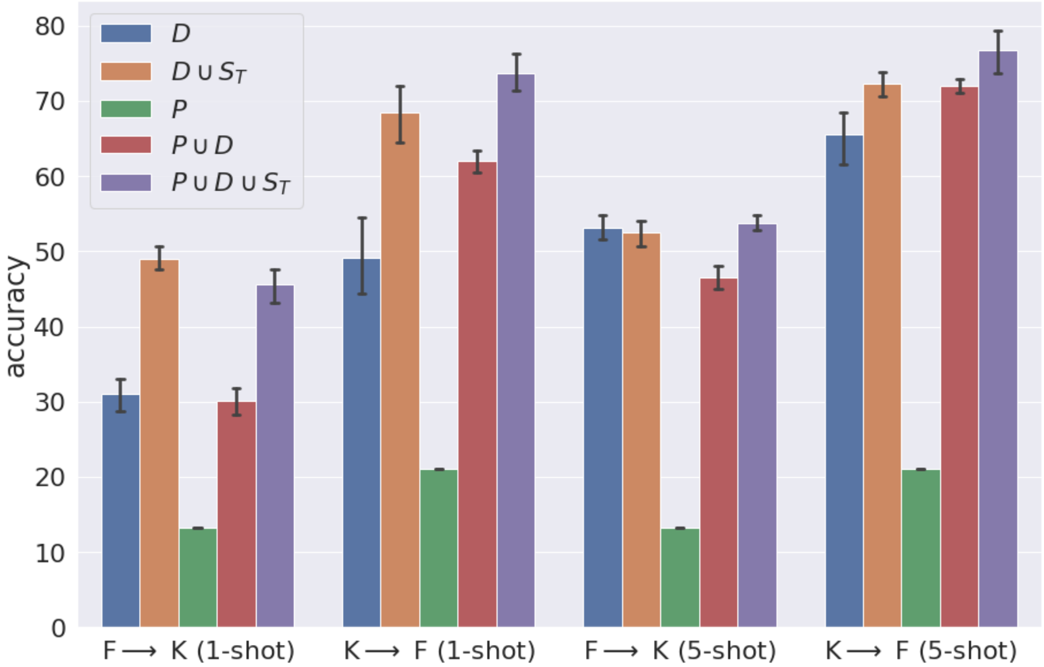

Figure 9: 1-shot and 5-shot transfer learning results between KMNIST and FashionMNIST datasets

B.6 Comparison with baseline

Comparison with Alvarez-Melis & Fusi (2021)'s approach. We adopted the same values of parameters (number of steps, step size) from the paper of Alvarez-Melis & Fusi (2021) and experimented with different values of entropy regularization λ. Entropy regularization λ is a hidden parameter in their code and they reported using the value of $\lambda = 100$ in transfer learning experiments. Also, we experimented with different methods in their code and found that "xyaugm" gives the best qualitative gradient flow results. We test their algorithm using the same transfer learning setting as ours (using the same clustered data as our experiments), see Table 2 and Table 3 for results.

Table 2: KMNIST → MNIST

Accuracy	$D \cup S_T$	$P \cup D \cup S_T$
$\lambda = 0.001$	0.1941	0.2478
$\lambda = 0.01$	0.1175	0.2998
$\lambda = 1.0$	0.1739	0.2951
$\lambda = 100$	0.2093	0.3025

Table 3: FMNIST → MNIST

Accuracy	$D \cup S_T$	$P \cup D \cup S_T$
$\lambda = 0.001$	0.4488	0.3083
$\lambda = 0.01$	0.3915	0.2897
$\lambda = 1.0$	0.5268	0.4627
$\lambda = 100$	0.2917	0.3307

For runtime comparison, the default device for Alvarez-Melis & Fusi (2021)'s code is on the CPU. As we run their code on the GPU, the kernel crashed without giving any informative errors. Thus, we will compare our codes' runtime per step on the CPU. While our approach takes about 11.72 seconds, the approach in Alvarez-Melis & Fusi (2021) requires 74.78 seconds.