Generating Synthetic Datasets by Interpolating along Generalized Geodesics

Jiaojiao Fan[†1] **David Alvarez-Melis**[2]

[1]Georgia Tech, Atlanta, Georgia, USA
[2]Microsoft Research & Harvard University, Cambridge, Massachusetts, USA

Abstract

Data for pretraining machine learning models often consists of collections of heterogeneous datasets. Although training on their union is reasonable in agnostic settings, it might be suboptimal when the target domain —where the model will ultimately be used— is known in advance. In that case, one would ideally pretrain only on the dataset(s) most similar to the target one. Instead of limiting this choice to those datasets already present in the pretraining collection, here we explore extending this search to all datasets that can be synthesized as 'combinations' of them. We define such combinations as multi-dataset interpolations, formalized through the notion of generalized geodesics from optimal transport (OT) theory. We compute these geodesics using a recent notion of distance between labeled datasets, and derive alternative interpolation schemes based on it: using either barycentric projections or optimal transport maps, the latter computed using recent neural OT methods. These methods are scalable, efficient, and —notably— can be used to interpolate even between datasets with distinct and unrelated label sets. Through various experiments in transfer learning in computer vision, we demonstrate this is a promising new approach for targeted on-demand dataset synthesis.

1 INTRODUCTION

Recent progress in machine learning has been characterized by the rapid adoption of large pretrained models as a fundamental building block [Brown et al., 2020]. These models are typically pretrained on large amounts of general-purpose data and then adapted (e.g., *fine-tuned*) to a specific task of interest. Such pretraining datasets usually draw from multiple heterogeneous data sources, e.g., arising from different domains or sources. Traditionally, all available datasets are used in their entirety during pretraining, for example by pooling them together into a single dataset (when they all share the same label sets) or by training in all of them sequentially one by one. These strategies, however, come with important disadvantages. Training on the union of multiple datasets might be prohibitive or too time-consuming, and it might even be detrimental — indeed, there is a growing line of research showing that removing pretraining data sometimes improves transfer performance [Jain et al., 2022]. On the other hand, sequential learning (i.e., consuming datasets one by one) is infamously prone to *catastrophic forgetting* [McCloskey and Cohen, 1989, Kirkpatrick et al., 2017]: the information from earlier datasets gradually vanishing as the model is trained on new datasets. The pitfalls of both of these approaches suggest training instead on a *subset* of the available pretraining datasets, but how to choose that subset is unclear. However, when the target dataset on which the model is to be used is known in advance, the answer is much easier: intuitively, one would train only of those relevant to the target one: e.g., those most similar to it. Indeed, recent work has shown that selecting pretraining datasets based on the distance to the target is a successful strategy [Alvarez-Melis and Fusi, 2020, Gao and Chaudhari, 2021]. However, such methods are limited to selecting (only) among individual datasets already present in the collection.

In this work, we propose a novel approach to *generate* synthetic pretraining datasets as combinations of existing ones. In particular, this method searches among all possible continuous combinations of the available datasets and thus is not limited to selecting specifically one of them. When given access to the target dataset of interest, we seek among all such combinations the one closest (in terms of a metric between datasets) to the target. By characterizing datasets as sampled from an underlying probability distribution, this problem can be understood as a generalization (from Euclidean to probability space) of the problem of finding among the con-

[†]Work done partly during an internship at Microsoft Research.

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

vex hull of a set of reference points, that which is closest to a query point. While this problem has a simple closed-form solution in Euclidean space (via an orthogonal projection), solving it in probability space is —as we shall see here— significantly more challenging.

We tackle this problem from the perspective of interpolation. Formally, we model the combination of datasets as an interpolation between their distributions, formalized through the notion of geodesics in probability space endowed with the Wasserstein metric [Ambrosio et al., 2008, Villani, 2008]. In particular, we rely on *generalized geodesics* [Craig, 2016, Ambrosio et al., 2008]: constant-speed curves connecting a pair (or more) distributions parametrized with respect to a 'base' distribution, whose role is played by the target dataset in our setting. Computing such geodesics requires access to either an optimal transport coupling or a map between the base distribution and every other reference distribution. The former can be computed very efficiently with off-the-shelf OT solvers, but are limited to generating only as many samples as the problem is originally solved on. In contrast, OT maps allow for on-demand out-of-sample mapping and can be estimated using recent advances in neural OT methods [Fan et al., 2020, Korotin et al., 2022b, Makkuva et al., 2020]. However, most existing OT methods assume unlabeled (feature-only) distributions, but our goal here is to interpolate between classification (i.e., labeled) datasets. Therefore, we leverage a recent generalization of OT for labeled datasets to compute couplings [Alvarez-Melis and Fusi, 2020] and adapt and generalize neural OT methods to the labeled setting to estimate OT maps.

In summary, the contributions of this paper are: (i) a novel approach to generate new synthetic classification datasets from existing ones by using geodesic interpolations, applicable even if they have disjoint label sets, (ii) two efficient methods to solve OT between labeled datasets, which might be of independent interest, (iii) empirical validation of the method in various transfer learning settings.

2 RELATED WORK

Mixup and related In-Domain Interpolation Generating training data through convex combinations was popularized by *mixup* [Zhang et al., 2018]: a simple data augmentation technique that interpolates features and labels between pairs of points. This and other works based on it [Zhang et al., 2021, Chuang and Mroueh, 2021, Yao et al., 2021] use mixup to improve in-domain model robustness [Zhu et al., 2023] and generalization by increasing the in-distribution diversity of the training data. Although sharing some intuitive principles with mixup, our method interpolates entire datasets —rather than individual datapoints— with the goal of improving across-distribution diversity and therefore out-of-domain generalization.

Dataset synthesis in machine learning Generating data beyond what is provided as a training dataset is a crucial component of machine learning in practice. Basic transformations such as rotations, cropping, and pixel transformations can be found in most state-of-the-art computer vision models. More recently, Generative Adversarial Nets (GAN) have been used to generate synthetic data in various contexts [Bowles et al., 2018, Yoon et al., 2019], a technique that has proven particularly successful in the medical imaging domain [Sandfort et al., 2019]. Since GANs are trained to replicate the dataset on which they are trained, these approaches are typically confined to generating in-distribution diversity and typically operate on features only.

Discrete OT, Neural OT, Gradient Flows Barycentric projection [Ambrosio et al., 2008, Perrot et al., 2016] is a simple and effective method to approximate an optimal transport map with discrete regularized OT. On the other hand, there has been remarkable recent progress in methods to estimate OT maps in Euclidean space using neural networks [Makkuva et al., 2020, Fan et al., 2021, Rout et al., 2022], which have been successfully used for image generation [Rout et al., 2022], style transfer [Korotin et al., 2022b], among other applications. However, the estimation of an optimal map between (labeled) datasets has so far received much less attention. Some conditional Monge map solvers [Bunne et al., 2022a] utilize the label information in a semi-supervised manner, where they assume the label-to-label correspondence between two distributions is known. Our method differs from theirs in that we do not require a pre-specified label-to-label mapping, but instead estimate it from data. Geodesics and interpolation in general metric spaces have been studied extensively in the optimal transport and metric geometry literature [McCann, 1997, Agueh and Carlier, 2011, Ambrosio et al., 2008, Santambrogio, 2015, Villani, 2008, Craig, 2016], albeit mostly in a theoretical setting. Gradient flows [Santambrogio, 2015], increasingly popular in machine learning to model existing processes [Bunne et al., 2022b, Mokrov et al., 2021, Fan et al., 2022, Hua et al., 2023] or solving optimization problems over datasets [Alvarez-Melis and Fusi, 2021], provide an alternative approach for interpolation between distributions but are computationally expensive.

3 BACKGROUND

3.1 DISTRIBUTION INTERPOLATION WITH OT

Consider $\mathcal{P}(\mathcal{X})$ the space of probability distributions with finite second moments over some Euclidean space $\mathcal{X}$. Given $\mu, \nu \in \mathcal{P}(\mathcal{X})$, the Monge formulation optimal transport problem seeks a map $T : \mathcal{X} \to \mathcal{X}$ that transforms μ into ν at a minimal cost. Formally, the objective of this problem is $\min_{T:T\sharp\mu=\nu} \int_{\mathbb{R}^d} \|x - T(x)\|_2^2 \mathrm{d}\mu(x)$, where the minimization is over all the maps that pushforward distribution μ into distribution ν. While a solution to this problem might not exist, a relaxation due to Kantorovich is guaranteed to have one. This modified version yields the Wasserstein-2 distance: $W_2^2(\mu,\nu) = \min_{\pi\in\Pi(\mu,\nu)} \int_{\mathbb{R}^d} \|x - x'\|_2^2 \mathrm{d}\pi(x,x')$, where now the constraint set $\Pi(\mu,\nu) = \{\pi \in \mathcal{P}(\mathcal{X}^2) \mid P_{0\sharp}\pi = \mu, P_{1\sharp\pi} = \nu\}$ contains all couplings with marginals μ and ν. The optimal such coupling is known as the OT plan. A celebrated result by Brenier [1991] states that whenever P has density with respect to the Lebesgue measure, the optimal T^* exists and is unique. In that case, the Kantorovich and Monge formulations coincide and their solutions are linked by $\pi^* = (\mathrm{Id}, T^*)_\sharp\mu$ where Id is the identity map. The Wasserstein-2 distance enjoys many desirable geometrical properties compared to other distances for distributions [Ambrosio et al., 2008]. One such property is the characterization of geodesics in probability space [Agueh and Carlier, 2011, Santambrogio, 2015]. When $\mathcal{P}(\mathcal{X})$ is equipped with metric W_2, the unique minimal geodesic between any two distributions μ_0 and μ_1 is fully determined by π, the optimal transport plan between them, through the relation:

$$\rho_t^D := ((1-t)x + ty)\sharp\pi(x,y), \quad t \in [0,1],$$

known as *displacement interpolation*. If the Monge map from μ_0 to μ_1 exists, the geodesic can also be written as

$$\rho_t^M := ((1-t)\mathrm{Id} + tT^*)\sharp\mu_0, \quad t \in [0,1],$$

and is known as *McCann's interpolation* [McCann, 1997]. It is easy to see that $\rho_0^M = \mu_0$ and $\rho_1^M = \mu_1$.

Such interpolations are only defined between two distributions. When there are $m \geq 2$ marginal distributions $\{\mu_1, \dots, \mu_m\}$, the *Wasserstein barycenter*

$$\rho_a^B := \arg\min_\rho \sum_{i=1}^m a_i W_2^2(\rho, \mu_i), \quad a \in \Delta_{m-1} \subset \mathbb{R}^m_{\geq 0}$$

generalizes McCann's interpolation [Agueh and Carlier, 2011]. Intuitively, the interpolation parameters $a = [a_1, \dots, a_m]$ determine the 'mixture proportions' of each dataset in the combination, akin to the weights in a convex combination of points in Euclidean space. In particular, when a is a one-hot vector with $a_i = 1$, then $\rho_a^B = \mu_i$, i.e., the barycenter is simply the i-th distribution. Barycenters have attracted significant attention in machine learning recently [Srivastava et al., 2018, Korotin et al., 2021], but they remain challenging to compute in high dimension [Fan et al., 2020, Korotin et al., 2022a].

Another limitation of these interpolation notions is the non-convexity of W_2^2 along them. In Euclidean space, given three points $x_1, x_2, y \in \mathbb{R}^d$, the function $t \mapsto \|x_t - y\|_2^2$, where x_t is the interpolation $x_t = (1-t)x_1 + tx_2$, is convex. In contrast, in Wasserstein space, neither the function $t \mapsto W_2^2(\rho_t^M, \nu)$ or $a \mapsto W_2^2(\rho_a^B, \nu)$ are guaranteed to be convex [Santambrogio, 2017, §4.4]. This complicates their theoretical analysis, such as in gradient flows. To circumvent this issue, Ambrosio et al. [2008] introduced the *generalized geodesic* of $\{\mu_1, \dots, \mu_m\}$ with base $\nu \in \mathcal{P}(\mathcal{X})$:

$$\rho_a^G := \left(\sum_{i=1}^m a_i T_i^*\right)\sharp\nu, \quad a \in \Delta_{m-1},$$

where T_i^* is the optimal map from ν to μ_i.

Lemma 1. *The functional $\mu \mapsto W_2^2(\mu,\nu)$ is convex along the generalized geodesics, and $W_2^2(\rho_a^G, \nu) \leq \sum_{i=1}^m a_i W_2^2(\mu_i, \nu)$.*

Thus, unlike the barycenter, the generalized geodesic does yield a notion of convexity satisfied by the Wasserstein distance and is easier to compute. The proof of Lemma 1 is postponed to §A. For these reasons, we adopt this notion of interpolation for our approach. It remains to discuss how to use it on (labeled) datasets.

3.2 DATASET DISTANCE

Consider a dataset $\mathcal{D}_P = \{z^{(i)}\}_{i=1}^N = \{x^{(i)}, y^{(i)}\}_{i=1}^N \overset{i.i.d.}{\sim} P(x,y)$. The Optimal Transport Dataset Distance (OTDD) [Alvarez-Melis and Fusi, 2020] measures its distance to another dataset $\mathcal{D}_Q$ as:

$$d_{\mathrm{OT}}^2(\mathcal{D}_P, \mathcal{D}_Q) = \min_{\pi\in\Pi(P,Q)} \int \left(\|x - x'\|_2^2 + W_2^2(\alpha_y, \alpha_{y'})\right) \mathrm{d}\pi(z, z'), \quad (1)$$

which defines a proper metric between datasets. Here, $\alpha_y, \alpha_{y'}$ are class-conditional measures corresponding to $P(x|y)$ and $Q(x|y')$. This distance is strongly correlated with transfer learning performance, i.e., the accuracy achieved when training a model on $\mathcal{D}_P$ and then fine-tuning and evaluating on $\mathcal{D}_P$. Therefore, it can be used to select pretraining datasets for a given target domain. Henceforth we abuse the notation P to represent both a dataset and its underlying distribution for simplicity. To avoid confusion, we use ν and μ to represent distributions in the feature space (typically $\mathbb{R}^d$) and use P, Q to represent distributions in the product space of features and labels.

4 DATASET INTERPOLATION ALONG GENERALIZED GEODESIC

Our method consists of two steps: estimating optimal transport maps between the target dataset and all training datasets (§4.1), and using them to generate a convex combination of these datasets by interpolating along generalized geodesics (§4.2). The OT map estimation is in feature space or original space depending on the dataset's dimension. For some downstream applications, we will additionally project the target dataset into the 'convex hull' of the training datasets (§4.3).

4.1 ESTIMATING OPTIMAL MAPS BETWEEN LABELED DATASETS

The OTDD is a special case of Wasserstein distance, so it is natural to consider the alternative Monge (map-based) formulation to (1). We propose two methods to approximate the OTDD map, one using the entropy-regularized OT and another one based on neural OT.

OTDD barycentric projection. Barycentric projections [Ambrosio et al., 2008, Pooladian and Niles-Weed, 2021] can be efficiently computed for entropic regularized OT using the Sinkhorn algorithm [Sinkhorn, 1967]. Assume that we have empirical distributions $\nu = \sum_{i=1}^{N_\nu} \frac{1}{N_\nu} \delta_{x_\nu^{(i)}}$ and $\mu = \sum_{i=1}^{N_\mu} \frac{1}{N_\mu} \delta_{x_\mu^{(i)}}$, where δ_x is the Dirac function at x. Denote all the samples compactly as matrices: $X_\nu = \left(x_\nu^{(1)}, \dots, x_\nu^{(N_\nu)}\right) \in \mathbb{R}^{N_\nu \times d}, X_\mu = \left(x_\mu^{(1)}, \dots, x_\mu^{(N_\mu)}\right) \in \mathbb{R}^{N_\mu \times d}$. After solving the optimal coupling $\pi^* := \min_{\pi \in \Pi(\nu,\mu)} \sum_{i,j} \|x_\nu^{(i)} - x_\mu^{(j)}\|^2 \pi(i,j)$, the barycentric projection can be expressed as $T_B(X_\nu) = N_\nu \pi^* X_\mu$. We extend this method to two datasets $Z_Q = \{X_Q, Y_Q\}, Z_P = \{X_P, Y_P\}$, where we have additional one-hot label data $Y_Q = (y_Q^{(1)}, \dots, y_Q^{(N_Q)}) \in \{0,1\}^{N_Q \times C_Q}, Y_P = (y_P^{(1)}, \dots, y_P^{(N_P)}) \in \{0,1\}^{N_P \times C_P}$. C_Q and C_P are the number of classes in dataset Q and P. We solve the optimal coupling $\pi^* \in \mathbb{R}_{\geq 0}^{N_P \times N_Q}$ for OTDD (1) following the regularized scheme in Alvarez-Melis and Fusi [2020]. The barycentric projection can then be written as:

$$\mathcal{T}_B(Z_Q) = [N_Q \pi^* X_P, N_Q \pi^* Y_P]. \tag{2}$$

The visualization of barycentric projected data appears in Figure 1. However, this approach has two important limitations: it can not naturally map out-of-sample data and it does not scale well to large datasets (due to the quadratic dependency on sample size). To relieve the scaling issue, we will use batchified version of OTDD barycentric projection in this paper (see complexity discussion in §6).

OTDD neural map. Inspired by recent approaches to estimate Monge maps using neural networks [Rout et al., 2022, Fan et al., 2021], we design a similar framework for the OTDD setting. Fan et al. [2021] approach the Monge OT problem with general cost functions by solving its max-min dual problem

$$\sup_f \inf_T \int [c(x, T(x)) - f(T(x))] \, d\nu(x) + \int f(x') d\mu(x').$$

We extend this method to the distributions involving labels by introducing an additional classifier in the map. Given two datasets P, Q, we parameterize the map $\mathcal{T}_N : \mathbb{R}^d \times [0,1]^{C_Q} \to \mathbb{R}^d \times [0,1]^{C_P}$ as

$$\mathcal{T}_N(z) = \mathcal{T}_N(x,y) = [\bar{x}; \bar{y}] = [G(z); \ell(G(z))],$$

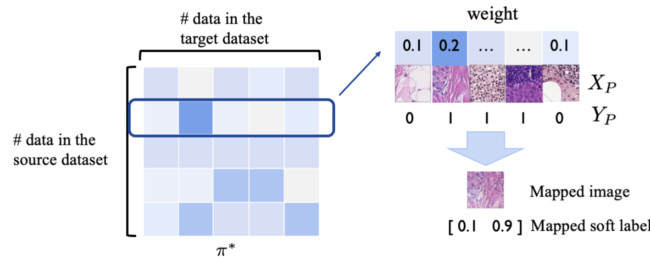


Figure 1: Visualization of OTDD barycentric projection on binary PCAM dataset. We first solve the optimal coupling $\pi^* \in [0,1]^{N_Q \times N_P}$ for the problem (1) using entropy regularization. Next, we map the i-th datapoint in the source dataset to a pair consisting of a weighted image and a weighted soft label. The weight vector, extracted from the i-th row of the coupling π^*, is then normalized to sum to 1. As a result, the mapped image (or soft label) is obtained as a convex combination of all the images (or one-hot labels) in the target dataset.

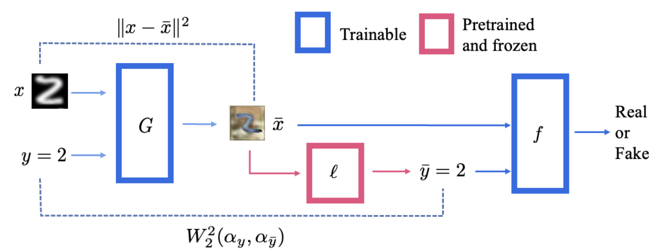


Figure 2: Training paradigm for learning the OTDD neural map betweem two datasets (distributions), parametrized via a pushforward feature map G and a labeling function ℓ, using a dual potential f.

where $G : \mathbb{R}^d \times [0,1]^{C_Q} \to \mathbb{R}^d$ is the pushforward feature map, and the $\ell : \mathbb{R}^d \to [0,1]^{C_P}$ is a frozen classifier that is pre-trained on the dataset P. Notice that, with the cost $c(z, \mathcal{T}(z)) = \|x - G(z)\|_2^2 + W_2^2(\alpha_y, \alpha_{\bar{y}})$, the Monge formulation of OTDD (1) reads $\inf_{\mathcal{T}\sharp Q = P} \int \|x - G(z)\|_2^2 + W_2^2(\alpha_y, \alpha_{\bar{y}}) dQ(z)$. We therefore propose to solve the max-min dual problem

$$\sup_f \inf_G \int \left[\|x - G(z)\|_2^2 + W_2^2(\alpha_y, \alpha_{\bar{y}})\right] dQ(z) - \int f(\bar{x}, \bar{y}) dQ(z) + \int f(x', y') dP(z').$$

Implementation details are provided in §B. Compared to previous conditional Monge map solvers [Bunne et al., 2022a, Asadulaev et al., 2022], the two methods proposed here: (i) do not assume class overlap across datasets, allowing for maps between datasets with different label sets; (ii) are invariant to class permutation and re-labeling; (iii) do not force one-to-one class alignments, e.g., samples can be mapped across similar classes.

4.2 CONVEX COMBINATION IN DATASET SPACE

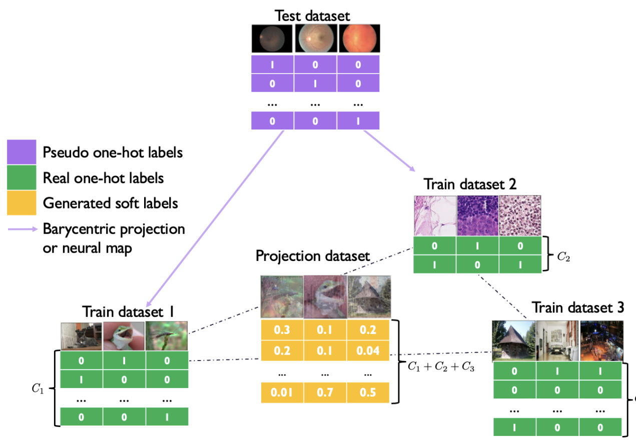


Figure 3: In few-shot settings, we use pseudo-labels for the test dataset, generated e.g. via kNN using the few-shot examples. If more labeled data from the test dataset is available, we use it instead of the pseudo-labels. The projection dataset has the same number of samples as the test dataset.

Computing generalized geodesics requires constructing convex combinations of datapoints from different datasets. Given a weight vector $a \in \Delta_{m-1}$, features can be naturally combined as $x_a = \sum_{i=1}^m a_i x_i$. But combining labels is not as simple because: (i) we allow for datasets with a different number of labels, so adding them directly is not possible; (ii) we do not assume different datasets have the same label sets, e.g. MNIST (digits) vs CIFAR10 (objects). Our solution is to represent all labels in the same dimensional space by padding them with zeros in all entries corresponding to other datasets. As an example, consider three datasets with 2, 3, and 4 classes respectively. Given a label vector $y_1 \in \mathbb{R}^3$ for the first one, we embed it into $\mathbb{R}^9$ as $\tilde{y}_1 = [y_1; \mathbf{0}_3; \mathbf{0}_4]^\top$. Defining $\tilde{y}_2, \tilde{y}_3$ analogously, we compute their combination as $y_a = a_1\tilde{y}_1 + a_2\tilde{y}_2 + a_3\tilde{y}_3$. This representation is lossless and preserves the distinction of labels across datasets. The visualization of our convex combination is in Figure 3.

4.3 PROJECTION ONTO GENERALIZED GEODESIC OF DATASETS

We now put together the components in Sec 4.1 and 4.2 to construct generalized geodesics between datasets in two steps. First, we compute OTDD maps $\mathcal{T}_i^*$ between Q and all other datasets $P_i, i = 1, \dots, m$ using the discrete or neural OT approaches. Then, for any interpolation vector $a \in \Delta_{m-1}$ we identify a dataset along the generalized geodesic via

$$P_a := \left(\sum_{i=1}^m a_i \mathcal{T}_i^*\right) \sharp Q.$$

By using the convex combination method in §4.2 for labeled data, we can efficiently sample from P_a.

Next, we find the dataset P_a^* that minimizes the distance between P_a and Q, i.e. the projection of Q onto the generalized geodesic. We first approach this problem from a Euclidean viewpoint. Suppose there are several distributions $\{\mu_i\}_{i=1}^m$ and an additional distribution ν on Euclidean space $\mathbb{R}^d$. Lemma 1 guarantees there exists a unique parameter a^* that minimizes $W_2^2(\rho_a^G, \nu)$. However, finding a^* is far from trivial because there is no closed-form formula of the map $a \mapsto W_2^2(\rho_a^G, \nu)$ and it can be expensive to calculate $W_2^2(\rho_a^G, \nu)$ for all possible a. To solve this problem, we resort to another transport distance: the (2,ν)-transport metric.

Definition 1 (Craig [2016]). *Given distributions μ_i, μ_j, the (2,ν)-transport metric between them is given by*

$$W_{2,\nu}(\mu_i, \mu_j) := \left(\int \|T_i^*(x) - T_j^*(x)\|_2^2 d\nu(x)\right)^{1/2},$$

where T_i^* is the optimal map from ν to μ_i.

When ν has a density with respect to Lebesgue measure $W_{2,\nu}$ is a valid metric [Craig, 2016, Prop. 1.15]. Moreover, we can derive a closed-form formula for the map $a \mapsto W_{2,\nu}^2(\rho_a^G, \nu)$.

Proposition 1. $W_{2,\nu}^2(\rho_a^G, \nu) = \sum_{i=1}^m a_i W_{2,\nu}^2(\mu_i, \nu) - \frac{1}{2}\sum_{i \neq j} a_i a_j W_{2,\nu}^2(\mu_i, \mu_j)$.

This equation implies that given distributions $\{\mu_i\}, \nu$ in Euclidean space, we can trivially solve the optimal a^* that minimizes $W_{2,\nu}^2(\rho_a^G, \nu)$ by a quadratic programming solver*. The proof (§A) relies on Brenier's theorem. Inspired by this, we also define a transport metric for datasets:

Definition 2. *The squared (2,Q)-dataset distance is*

$$\mathcal{W}_{2,Q}^2(P_i, P_j) := \int \left(\|x_i - x_j\|_2^2 + W_2^2(\alpha_{y_i}, \alpha_{y_j})\right) dQ(z),$$

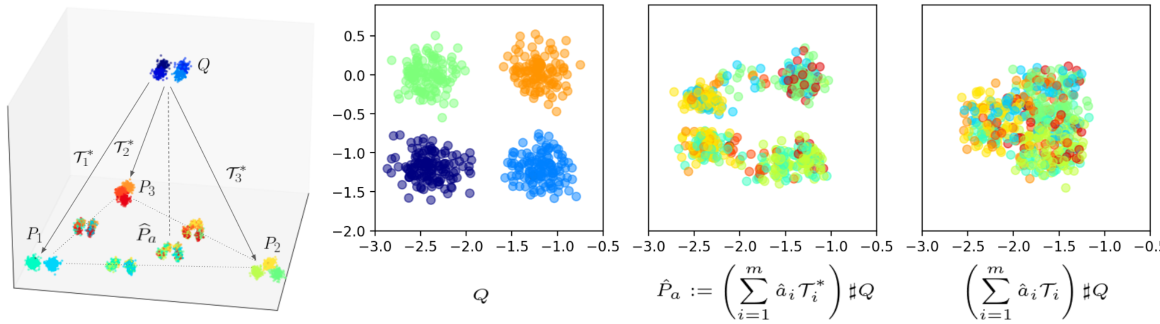
where $[x_i; y_i] = \mathcal{T}_i^*(z)$ and $\mathcal{T}_i^*$ is the OTDD map from Q to P_i.

Denote $\mathcal{P}_{2,Q}(\mathcal{X} \times \mathcal{P}(\mathcal{X}))$ as the set of all probability measures P that satisfy $d_{\mathrm{OT}}(P, Q) < \infty$ and the OTDD map from Q to P exists. The following result shows that (2,Q)-dataset distance is a proper distance. The proof is again deferred to §A.

Proposition 2. *$\mathcal{W}_{2,Q}$ is a valid metric on $\mathcal{P}_{2,Q}(\mathcal{X} \times \mathcal{P}(\mathcal{X}))$.*

Unfortunately, in this case $\mathcal{W}_{2,Q}^2(P_a, Q)$ does not have an analytic form like before because Brenier's theorem may

*We use the implementation https://github.com/stephane-caron/qpsolvers



(a) (b)

Figure 4: **Visualization and comparison of dataset interpolation methods.** (a) The reference dataset Q (with color-coded classes) is projected onto the generalized geodesic of the training datasets P_i, resulting in $\widehat{P}_a$. (b) 2D visualizations of (left-to-right): dataset Q, the 'optimal' interpolated dataset $\hat{P}_a := (\sum_{i=1}^{m} \hat{a}_i \mathcal{T}_i^*) \sharp Q$ using the true OTDD maps $\mathcal{T}_i^*$, and a naively interpolated dataset $(\sum_{i=1}^{m} \hat{a}_i \mathcal{T}_i) \sharp Q$ using randomly generated maps $\mathcal{T}_i$.

not hold for a general transport cost problem. However, we still borrow this idea and define an approximated projection $\widehat{P}_a$ as the minimizer of function

$$\mathcal{W}^2(P_a, Q) := \sum_{i=1}^{m} a_i \mathcal{W}^2_{2,Q}(P_i, Q) - \frac{1}{2} \sum_{i \neq j} a_i a_j \mathcal{W}^2_{2,Q}(P_i, P_j), \quad (3)$$

which is an analog of Proposition 1. Since P_a is defined by its interpolation weight a, solving $\widehat{P}_a$ is equivalent to finding a weight

$$\hat{a} = \arg\min_{a \in \Delta_{m-1}} \mathcal{W}^2(P_a, Q), \quad (4)$$

which is a simple quadratic programming problem. Unlike the Wasserstein distance, $\mathcal{W}^2_{2,Q}(\cdot, \cdot)$ is easier to compute because it does not involve optimization, so it is relatively cheap to find the minimizer of $\mathcal{W}^2(P_a, Q)$. Experimentally, we observe that $W^2_{2,Q}(P_a, Q)$ is predictive of model transferability across tasks. Figure 4(a) illustrates this projection on toy 3D datasets, color-coded by class.

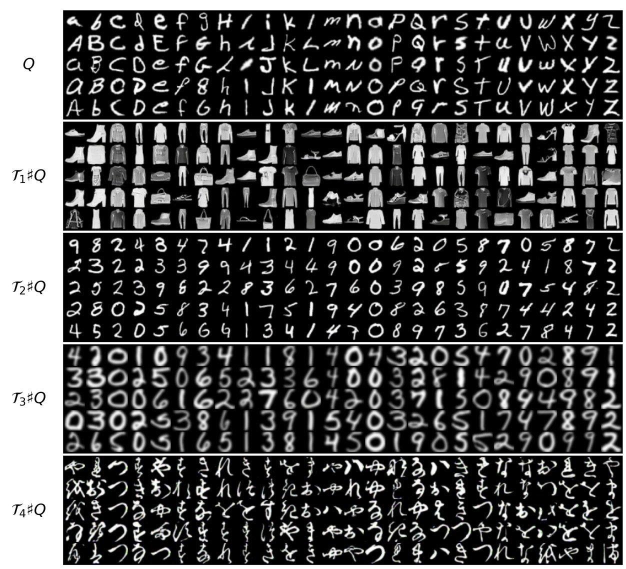


Figure 5: Datasets generated by pushing forward Q (the EMNIST dataset) towards Fashion-MNIST, MNIST, USPS, KMNIST, using OTDD maps $\mathcal{T}_i$, obtained using the neural OT method described in Section 4.1.

5 EXPERIMENTS

5.1 LEARNING OTDD MAPS

In this section, we visualize the quality of the learnt OTDD maps on both synthetic and realistic datasets.

Synthetic datasets Figure 4 (b) illustrates the role of the optimal map in estimating the projection of a dataset into the generalized geodesic hull of three others. Using maps $\mathcal{T}_i^*$ estimated via barycentric projection (2) results in better preservation of the four-mode class structure, whereas using non-optimal maps $\mathcal{T}_i$ based on random couplings (as the usual *mixup* does) destroys the class structure.

***NIST datasets** In Figure 5, we provide qualitative results of OTDD map from EMNIST (letter) [Cohen et al., 2017] dataset to all other *NIST dataset and USPS dataset. At this point, we can confirm three traits of OTDD map, which are mentioned at the end of §4.1.

1) We don't assume a known source label to target label correspondence. So we can map between two irrelevant datasets such as EMNIST and FashionMNIST. 2) The map is invariant to the permutation of label assignment. For example, we show two different labelling in Figure 1 in appendix and the final OTDD map will be the same. 3) It doesn't enforce the label-to-label mapping but would follow the fea-

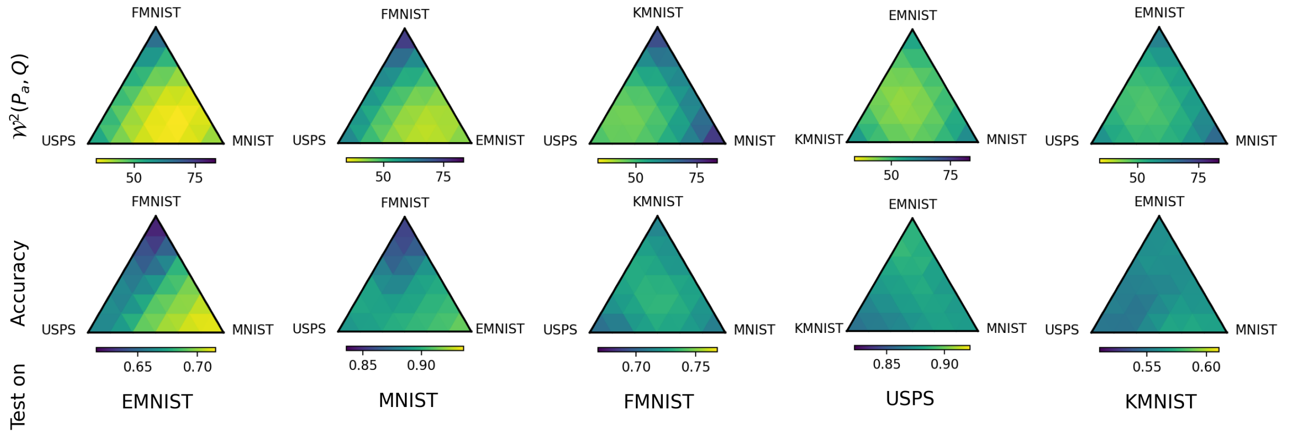


Figure 6: Relationship between the function $\mathcal{W}^2(P_a, Q)$ and the accuracy of the fine-tuned model. The model trained on the projection dataset $\hat{P}_a$, i.e. the minimizer of $\mathcal{W}^2(P_a, Q)$, tends to have a better generalization accuracy. The training datasets are marked on the vertexes of each ternary plot. Each ternary plot is an average of 5 runs with distinct random seeds.

ture similarity. From Figure 5, we notice many cross-class mapping behaviors. For example, when the target domain is USPS [Hull, 1994] dataset, the lower-case letter "l" is always mapped to digit 1, and the capital letter "L" is mapped to other digits such as 6 or 0 because the map follows the feature similarity.

5.2 TRANSFER LEARNING ON *NIST DATASETS

Next, we use our framework to generate new pretraining datasets for transfer learning. Preceding works illustrate that the transfer learning performance can be quite sensitive to the type of test datasets if there is abundant training data from the test task [Zhai et al., 2019, Table 1]. Thus, we will focus on the few-shot setting, where we only have few labeled data from the test task. We first show that the generalization ability of training models has a strong correlation with the distance $\mathcal{W}^2_{2,Q}(P_a, Q)$. Then we compare our framework with several baseline methods.

Setup Given m labeled pretraining datasets $\{P_i\}$, we consider a few-shot task in which only a limited amount of data from the target domain is labeled, e.g. 5 samples per class. The goal is to find a single dataset of size comparable to any individual P_i that yields the best generalization to the target domain when pre-training a model on it and fine-tuning on the target few-shot data. Here, we seek this training dataset within those generated by generalized geodesics $\{P_a\}$, which can be understood as weighted interpolations of the training datasets $\{P_i\}$. Note this includes individual datasets as particular cases when a is a one-hot vector.

Connection to generalization The closed-form expression of $W^2_{2,\nu}(\rho_a^G, \nu)$ (Prop. 1) provides a distance between a base distribution ν and the distribution along generalized geodesic ρ_a^G in Euclidean space. We study its analog (3) for labeled datasets Q and $\{P_i\}$ and visualize it in Figure 6 (first row). To investigate the generalization abilities of models trained on different datasets, we discretize the simplex Δ_2 to obtain 36 interpolation parameters a, and train a 5-layer LeNet classifier on each P_a. Then we fine-tune all of these classifiers on the few-shot test dataset Q with only 20 samples per each class. We control the same number of training iterations and fine-tuning iterations across all experiments. The second row of Figure 6 shows fine-tuning accuracy. Comparing the first row and the second, we find the accuracy and $\mathcal{W}^2(P_a, Q)$ are highly correlated. This implies that the model trained on the minimizer dataset of $\mathcal{W}^2(P_a, Q)$ tends to have a better generalization ability. We fix the same colorbar range for all heatmaps across datasets to highlight the impact of training dataset choice. A more concrete visualization of the correlation between $\mathcal{W}^2(P_a, Q)$ and accuracy is shown in Figure 5 in appendix.

For some test datasets, the choice of training dataset strongly affects the fine-tuning accuracy. For example, when Q is EMNIST and the training dataset is FMNIST, the fine-tuning accuracy is only $\sim 60\%$, but this can be improved to $> 70\%$ by choosing an interpolated dataset closer to MNIST. This is reasonable because MNIST is more similar to EMNSIT than FMNIST or USPS. To some test datasets like FMNIST and KMNIST, this difference is not so obvious because all training datasets are all far away from the test dataset.

Comparison with baselines. Next, we compare our method with several baseline methods on NIST datasets. In each set of experiments, we select one *NIST dataset as the target domain, and use the rest for pre-training. We consider a 5-shot task, so we **randomly** choose 5 samples per class to be the labeled data, and treat the remaining samples as unlabeled. Our method first trains a model on $\widehat{P}_a$,

Table 1: **Pretraining on synthetic data**. For each of the *NIST datasets, we treat it as the target domain and pretrain a neural net on a synthetic dataset generated as a combination of the remaining dataset with three interpolation methods. Here we show 5-shot transfer accuracy (mean $\pm$ s.d. over 5 runs). The first baseline is to create a synthetic dataset as a training dataset by Mixup among datasets. For Mixup, we randomly sample data from each training dataset, and do the convex combination of them with weight $\hat{a}$ (see Eq. (4)). We use the same convex combination method in §4.2, thus this Mixup baseline is equivalent to our framework with suboptimal OTDD maps. The other two baselines (the bottom block) skip the transfer learning part, and directly train the model or solve 1-NN on the few-shot test dataset.

Methods	MNIST-M	EMNIST	MNIST	FMNIST	USPS	KMNIST
OTDD barycentric projection	**42.10±4.37**	**67.06±2.55**	**93.74±1.46**	**70.12±3.02**	86.01±1.50	**52.55±2.73**
OTDD neural map	40.06±4.75	65.32±1.80	88.78±3.85	70.02±2.59	83.80±1.60	50.32±3.10
Mixup with weights $\hat{a}$	33.85±2.22	60.95±1.38	88.68±1.57	66.74±3.79	**88.61±2.00**	48.16±3.38
Train on few-shot dataset	19.10±3.57	53.60±1.18	72.80±3.10	60.50±3.07	80.73±2.07	41.67±2.11
1-NN on few-shot dataset	20.95±1.39	39.70±0.57	64.50±3.32	60.92±2.42	73.64±2.35	40.18±3.09

and fine-tunes the model on the 5-shot target data. To obtain $\widehat{P}_a$, we use barycentric projection or neural map to approximate the OTDD maps from the test to training datasets. Our results are shown in the first two rows in Table 1. Overall, transfer learning can bring additional knowledge from other domains and improve the test accuracy by at most 21%. Among the methods in the first block, training on datasets generated by OTDD barycentric projection outperforms others except USPS dataset, where the difference is only about 2.6%.

5.3 TRANSFER LEARNING ON VTAB DATASETS

Finally, we use our method for transfer learning with large-scale VTAB datasets [Zhai et al., 2019]. In particular, we take Oxford-IIIT Pet dataset as the target domain, and use Caltech101, DTD, and Flowers102 for pre-training. To encode a richer geometry in our interpolation, we embed the datasets using a masked auto encoder (MAE) [He et al., 2022] and learn the OTDD map in this ($\sim$200K dimensional) latent space. Since OTDD barycentric projection consistently works better than OTDD neural map (see Table 1), we only use barycentric projection in this section. We use ResNet-18 as the model architecture and pre-train the model on decoded MAE images (interpolated dataset) or original images (single dataset). Meanwhile, Mixup baseline is over pixel space and therefore does not utilize embeddings at all.

The pre-training interpolation dataset generated by our method has 'optimal' mixture weights $a = (0.43, 0.24, 0.33)$ for (CALTECH101, DTD, FLOWERS102), suggesting a stronger similarity between the first of these and the target domain (PETS). This is consistent with the single-dataset transfer accuracies shown in Table 2. However, their interpolation yields better transfer than any single dataset, particularly when using our full method (interpolating using OTDD map with optimal mixture weights).

In Table 2, we compute relative improvement per run, and then average these across runs; in other words, we compute the mean of ratios (MoR) rather than the ratio of means (RoM). Our reasoning for doing this was (i) controlling for the 'hardness' inherent to the randomly sampled subsets of PET by relativizing before averaging and (ii) our observation that it is common practice to compute MoR when the denominator and numerator correspond to paired data (as is the case here), and the terms in the sum are sampled i.i.d. (again, satisfied in this case by the randomly sampled subsets of the target domain).

Table 2 shows a high deviation due to a particularly good result generated by the non-transfer learning baseline with seed 2, while other methods such as Caltech101 pretraining and Flowers102 pretraining had particularly bad results with the same seed.

6 CONCLUSION AND DISCUSSION

The method we introduce in this work provides, as shown by our experimental results, a promising new approach to generate synthetic datasets as combinations of existing ones. Crucially, our method allows one to combine datasets even if their label sets are different, and is grounded on principled and well-understood concepts from optimal transport theory. Two key applications of this approach that we envision are:

- **Pretraining data enrichment**. Given a collection of classification datasets, generate additional interpolated datasets to increase diversity, with the aim of achieving better out-of-distribution generalization. This could be done even without knowledge of the specific target domain (as we do here) by selecting various datasets to play the role of the 'reference' distribution.
- **On-demand optimized synthetic data generation**. Generate a synthetic dataset, by combining existing ones, that

Table 2: **Transfer Learning on VTAB datasets**. The table shows relative improvement (w.r.t. a no-transfer baseline) of test accuracy on OXFORD-IIIT PET (mean ± std over 5 runs) given only 1000 randomly selected samples of this dataset to fine-tune. The first three rows show single-pretraining-dataset baselines, and the remaining rows show methods that pretrain on a synthetic interpolation of these three, generated using Mixup or our proposed OTDD Map, using uniform or $\hat{a}$ (see Eq. (4)) dataset interpolation weights. The pooling baseline pretrains on a dataset including all the pre-training datasets. To construct the sub-pooling pretraining dataset, for each training sample from the target dataset (PET) we find its 10-nearest neighbors (in embedding space) from across all pretraining datasets, and label them as belonging to the class from the target domain.

Pre-Training	Map	Weights	Rel. Improv. (%)
CALTECH101	–	–	59.68 ± 41.44
DTD	–	–	-1.17 ± 9.52
FLOWERS102	–	–	-2.45 ± 26.25
Pooling	–	–	28.96 ± 18.29
Sub-pooling	–	–	3.00 ± 19.10
Interpolation	Mixup	uniform	33.26 ± 21.30
Interpolation	Mixup	$\hat{a}$	51.99 ± 34.10
Interpolation	OTDD	uniform	82.61 ± 25.93
Interpolation	OTDD	$\hat{a}$	**95.17 ± 20.57**

is 'optimized' for transferring a model to a new (data-limited) target domain.

Complexity The complexity of solving OTDD barycentric projection by Sinkhorn algorithm is $\mathcal{O}(N^2)$ [Dvurechensky et al., 2018], where N is the number of data in both datasets. This can be expensive for large-scale datasets. In practice, we solve the batched barycentric projection, i.e. take a batch from both datasets and solve the projection from source to target batch, and we normally fix batch size B as 10^4. This reduces the complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(BN)$. The complexity of solving OTDD neural map is $\mathcal{O}(BKH)$, where K is number of iterations, and H is the size of the network. We always choose $K = \mathcal{O}(N)$ in the experiments. The complexity of solving all the $(2, Q)$-dataset distances in (3) is $\mathcal{O}(m^2N)$ since we need to solve the dataset distance between each pair of training datasets. Putting these pieces together, the complexity of approximating the interpolation parameter $\hat{a}$ for the minimizer of (3) is $\mathcal{O}(N(B + m^2))$.

Memory As the number of pre-training tasks (m) increases, our method, which generates an interpolated label by concatenating labels from all tasks, creates an increasingly sparse vector. Consequently, the memory demands of the classifier's output layer, which is proportional to m, could rise significantly.

Barycentric projection vs Neural map These two versions of our method offer complementary advantages. While estimating the OT map allows for easy out-of-sample mapping and continuous generation, the barycentric projection approach often yields better downstream performance (Table 1). We hypothesize this is due to the barycentric projection relying on (re-weighted) *real* data, while the neural map *generates* data which might be noisy or imperfect.

Pixel space vs feature space We present results with OTDD mapping in both pixel space (§5.2) and feature space (§5.3). For the VTAB datasets with regular-sized images (e.g. $256 \times 256 \times 3$), we found that the feature space is more appropriate for measuring data distance. For small-scale images like NIST, feature space may be overkill because most foundation models are trained on images with a larger size. In our preliminary experiments with NIST datasets, we attempted a feature space approach using an off-the-shelf ResNet-18 model. However, we encountered challenges in achieving convergence when training OTDD neural maps with PyTorch ResNet-18 features.

High variance issue Our method is not limited to the data scarcity regime, but indeed this is the most interesting one from the transfer learning perspective, which is why we assume limited labeled data (but potentially much more unlabeled data) from the target domain distribution. This is a typical few-shot learning scenario. The quality of a learned OT map will likely depend on the number of samples used to fit it, and might suffer from high variance. To mitigate this in our setting, we opt for augmenting our dataset by generating additional pseudo-labeled data via kNN (Fig. 3). Recall that we do have access to more unlabeled data from the target domain, which is a common situation in practice.

Limitations Our method for generating a synthetic dataset relies on solving OTDD maps from the test dataset to each training dataset. These OTDD maps are tailored to the considered test dataset and can not be reused for a new test dataset. Another limitation is our framework is based on model training and fine-tuning pipeline. This can be resource-demanding for large-scale models, like GPT [Brown et al., 2020] or other similar models. Finally, if at least one of the datasets is imbalanced, our OTDD map will struggle to match the class with similar marginal distributions.

Acknowledgements

We thank Yongxin Chen and Nicolò Fusi for their invaluable comments, ideas, and feedback. We extend our gratitude to the anonymous reviewers for their useful feedback that significantly improved this work.

References

M. Agueh and G. Carlier. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2): 904–924, 2011.

D. Alvarez-Melis and N. Fusi. Geometric dataset distances via optimal transport. *Advances in Neural Information Processing Systems*, 33:21428–21439, 2020.

D. Alvarez-Melis and N. Fusi. Dataset dynamics via gradient flows in probability space. In *International Conference on Machine Learning*, pages 219–230. PMLR, 2021.

L. Ambrosio, N. Gigli, and G. Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

A. Asadulaev, A. Korotin, V. Egiazarian, and E. Burnaev. Neural optimal transport with general cost functionals. *arXiv preprint arXiv:2205.15403*, 2022.

C. Bowles, L. Chen, R. Guerrero, P. Bentley, R. Gunn, A. Hammers, D. A. Dickie, M. V. Hernández, J. Wardlaw, and D. Rueckert. GAN augmentation: Augmenting training data using generative adversarial networks, Oct. 2018.

Y. Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417, 1991.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are Few-Shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

C. Bunne, A. Krause, and M. Cuturi. Supervised training of conditional monge maps. *arXiv preprint arXiv:2206.14262*, 2022a.

C. Bunne, L. Papaxanthos, A. Krause, and M. Cuturi. Proximal optimal transport modeling of population dynamics. In G. Camps-Valls, F. J. R. Ruiz, and I. Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 6511–6528. PMLR, 2022b.

C.-Y. Chuang and Y. Mroueh. Fair mixup: Fairness via interpolation. In *International Conference on Learning Representations*, 2021.

G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

K. Craig. The exponential formula for the wasserstein metric. *ESAIM: Control, Optimisation and Calculus of Variations*, 22(1):169–187, 2016.

P. Dvurechensky, A. Gasnikov, and A. Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by sinkhorn's algorithm. In *International conference on machine learning*, pages 1367–1376. PMLR, 2018.

J. Fan, A. Taghvaei, and Y. Chen. Scalable computations of wasserstein barycenter via input convex neural networks. *arXiv preprint arXiv:2007.04462*, 2020.

J. Fan, S. Liu, S. Ma, H. Zhou, and Y. Chen. Neural monge map estimation and its applications. *arXiv preprint arXiv:2106.03812*, 2021.

J. Fan, Q. Zhang, A. Taghvaei, and Y. Chen. Variational Wasserstein gradient flow. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 6185–6215. PMLR, 2022.

Y. Gao and P. Chaudhari. An Information-Geometric distance on the space of tasks. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021.

K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

X. Hua, T. Nguyen, T. Le, J. Blanchet, and V. A. Nguyen. Dynamic flows on curved space generated by labeled data. *arXiv preprint arXiv:2302.00061*, 2023.

J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

S. Jain, H. Salman, A. Khaddaj, E. Wong, S. M. Park, and A. Madry. A data-based perspective on transfer learning. *arXiv preprint arXiv:2207.05739*, 2022.

J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *PNAS*, 114(13):3521–3526, mar 2017.

A. Korotin, L. Li, J. Solomon, and E. Burnaev. Continuous wasserstein-2 barycenter estimation without minimax optimization. *arXiv preprint arXiv:2102.01752*, 2021.

A. Korotin, V. Egiazarian, L. Li, and E. Burnaev. Wasserstein iterative networks for barycenter estimation. *arXiv preprint arXiv:2201.12245*, 2022a.

A. Korotin, D. Selikhanovych, and E. Burnaev. Neural optimal transport. *arXiv preprint arXiv:2201.12220*, 2022b.

A. Makkuva, A. Taghvaei, S. Oh, and J. Lee. Optimal transport mapping via input convex neural networks. In *International Conference on Machine Learning*, volume 37, 2020.

R. J. McCann. A convexity principle for interacting gases. *Advances in mathematics*, 128(1):153–179, 1997.

M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In G. H. Bower, editor, *Psychology of Learning and Motivation*, volume 24, pages 109–165. Academic Press, Jan. 1989. doi: 10.1016/S0079-7421(08)60536-8.

P. Mokrov, A. Korotin, L. Li, A. Genevay, J. M. Solomon, and E. Burnaev. Large-Scale wasserstein gradient flows. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 15243–15256. Curran Associates, Inc., 2021.

M. Perrot, N. Courty, R. Flamary, and A. Habrard. Mapping estimation for discrete optimal transport. *Advances in Neural Information Processing Systems*, 29, 2016.

A.-A. Pooladian and J. Niles-Weed. Entropic estimation of optimal transport maps. *arXiv preprint arXiv:2109.12004*, 2021.

L. Rout, A. Korotin, and E. Burnaev. Generative modeling with optimal transport maps. In *International Conference on Learning Representations*, 2022.

V. Sandfort, K. Yan, P. J. Pickhardt, and R. M. Summers. Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. *Sci. Rep.*, 9(1):16884, Nov. 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-52737-x.

F. Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

F. Santambrogio. {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bulletin of Mathematical Sciences*, 7(1):87–154, 2017.

R. Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967.

S. Srivastava, C. Li, and D. B. Dunson. Scalable bayes via barycenter in wasserstein space. *The Journal of Machine Learning Research*, 19(1):312–346, 2018.

C. Villani. *Optimal transport, Old and New*, volume 338. Springer Science & Business Media, 2008. ISBN 9783540710493.

H. Yao, L. Zhang, and C. Finn. Meta-learning with fewer tasks through task interpolation. *arXiv preprint arXiv:2106.02695*, 2021.

J. Yoon, J. Jordon, and M. van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019.

X. Zhai, J. Puigcerver, A. Kolesnikov, P. Ruyssen, C. Riquelme, M. Lucic, J. Djolonga, A. S. Pinto, M. Neumann, A. Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

L. Zhang, Z. Deng, K. Kawaguchi, A. Ghorbani, and J. Zou. How does mixup help with robustness and generalization? In *International Conference on Learning Representations*, 2021.

J. Zhu, J. Qiu, A. Guha, Z. Yang, X. Nguyen, B. Li, and D. Zhao. Interpolation for robust learning: Data augmentation on geodesics. *arXiv preprint arXiv:2302.02092*, 2023.