Generating Synthetic Datasets by Interpolating along Generalized Geodesics

Jiaojiao Fan*
Georgia Tech
jiaojiaofan@gatech.edu

David Alvarez-Melis
Microsoft Research
daalvare@microsoft.com

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, we demonstrate this promising new approach to 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., from 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 evidence that removing pretraining data sometimes helps transfer performance (Jain et al., 2022). On the other hand, sequential learning (e.g., consuming datasets one by one) is infamously prone to *catastrophic forgetting* (McCloskey & Cohen, 1989; Kirkpatrick et al., 2017): the information from earlier datasets gradually vanishing as the model is trained on new datasets. While all of these suggest that training on only some subset of the pretraining datasets, how to choose these 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 on 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 & Fusi, 2020; Gao & Chaudhari, 2021). However, such methods are limited to selecting (only) among individual datasets already present in the collection.

*Work done while at Microsoft Research.

NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research.

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 select one of them necessarily. 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 a underlying probability distribution, this problem can be understood as a generalization (from Euclidean to probability space) of the problem of finding among the convex hull of a set of reference points, that closest to a query point. While this problem has a simple closed-form solution in Euclidean space (via an orthogonal projection), solving it probability space is, as we shall see here, much 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 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 generate 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 to labeled datasets to compute couplings (Alvarez-Melis & 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 compute generalize geodesics, which might be of independent interest, (iii) empirical validation of the method in a transfer learning setting.

2 Background

2.1 Distributional interpolation with OT

Consider $P(X)$ the space of probability distribution finite second moments over some Euclidean space X. Given $\quad ; \quad 2\ P(X)$, the Monge formulation optimal transport problem seeks a map $T : X \ ! \ X$ that transforms $\quad$ into $\quad$ at minimal cost. Formally, the objective of this problem is $\min_{T:T]\ =\ } \int_{\mathbb{R}^d} kx \quad T(x)k_2^2 \mathrm{d}\ (x)$; where the minimization is over all the maps that pushforward distribution $\quad$ into distribution $\quad$. While a solution to this problem might not exist, a relaxation due to Kantorovich is guaranteed to have one. This modified version yields the 2-Wasserstein distance: $W_2^2(\ ;\ ) = \min_{\in\ (\ ;\ )} \int_{\mathbb{R}^d} kx \quad x'k_2^2 \mathrm{d}\ (x; x')$; where now the constraint set $(\ ;\ ) = f \quad 2\ P(X^2)\ j\ P_{0]} \quad = \ ; P_{1]} \quad = \quad g$ contains all couplings with marginals $\quad$ and $\quad$. 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 Lebesgue measure, the optimal T^* exists and is unique. In that case, the Kantorovich and Monge formulations coincide and their solutions are linked by $^* = (\mathrm{Id}; T^*)_]$ 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 & Carlier, 2011; Santambrogio, 2015). When $P(X)$ is equipped with metric W_p, the unique minimal geodesic between any two distributions $_0$ and $_1$ is fully determined by $\quad$, the optimal transport plan between them, through the relation:

$$\overset{D}{t} := ((1 \quad t)x + ty)] \ (x; y); \quad t\ 2\ [0; 1];$$

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

$$\overset{M}{t} := ((1 \quad t)\mathrm{Id} + tT^*)] \ _1; \quad t\ 2\ [0; 1]; \tag{1}$$

and is known as *McCann's interpolation* (McCann, 1997). It easy to see that $\overset{M}{0} = \ _1$ and $\overset{M}{1} = \ _2$.

Such interpolations are only defined between two distributions. When there are $m \quad 2$ marginal distributions $f \ _1; : : : ; \ _m g$, the *Wasserstein barycenter* $\overset{B}{a} := \arg\min \ \sum_{i=1}^m a_i W_2^2(\ ;\ _i); \quad a\ 2\ _m \quad \mathbb{R}^m$ generalizes McCann's interpolation (Agueh & Carlier, 2011). Intuitively, the interpolation

parameters $a = [a_1; \dots; a_m]$ determine the `mixture proportions' of each dataset in the combination, akin to a convex combination of points in Euclidean space. In particular, when a is a one-hot vector with $a_i = 1$, then $\bar{}^{B}_{a} = {}_i$, i.e., the barycenter is simply the ith distribution. Barycenters have attracted great 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 $x_1; x_2; y \in R^d$, 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 $t \mapsto W_2^2(\bar{}^{M}_{t}; \ )$ or $t \mapsto W_2^2(\bar{}^{B}_{a}; \ )$ are guaranteed to be convex (Santambrogio, 2017, Sec. 4.4). This complicates theoretical analysis, such as in gradient flows. To circumvent this issue, Ambrosio et al. (2008) introduced the generalized geodesics of $\{{}_1; \dots; {}_m\}$ with base $$ and defined it as $\bar{}^{G}_{a} := (\sum_{i=1}^{m} a_i T_i)_\# \ ; \ a \in {}^m$; where T_i is the optimal map from $$ to ${}_i$.

Lemma 1. The functional $\mapsto W_2^2(\ ; \ )$ is convex along the generalized geodesics, and $W_2^2(\bar{}^{G}_{a}; \ ) \sum_{i=1}^{m} a_i W_2^2({}_i; \ )$.

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

2.2 Dataset distance

Consider a dataset $D_P = \{z^{(i)}\}_{i=1}^{N} = \{x^{(i)}; y^{(i)}\}_{i=1}^{N} \overset{\text{i.i.d.}}{\sim} P(x; y)$. The Optimal Transport Dataset Distance (OTDD) (Alvarez-Melis & Fusi, 2020) measures its distance to another dataset D_Q:

$$d_{OT}^2(D_P; D_Q) = \min_{\in (P;Q)} \int \|x - x'\|_2^2 + W_2^2({}_y; {}_{y'}) \, d(z; z'); \qquad (2)$$

which defines a proper metric between datasets. Here, ${}_y, {}_{y'}$ 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 D_P and then fine-tuning and evaluating on 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 and to represent distributions in the feature space, which is Euclidean space, and use P, Q to represent distributions in the product space of features and labels.

3 Method and algorithm

Our method consists of two steps: estimating optimal transport maps between the target dataset and all training datasets (Sec. 3.1), and using them to generate a convex combinations of these datasets by interpolating along generalized geodesics (Sec. 3.2). For some downstream applications we will additionally project the target dataset into the `convex hull' of the training datasets (Sec. 3.3).

3.1 Solving optimal map between labelled datasets

The OTDD is a special case of Wasserstein distance, so it is natural to consider the alternative Monge (map-based) formulation to (2). 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 & Niles-Weed, 2021) can be efficiently computed for entropic regularized OT using the Sinkhorn algorithm (Sinkhorn, 1967). Assume that we have i.i.d. samples $X = (x^{(1)}; \dots; x^{(N)}); X = (x^{(1)}; \dots; x^{(N)})$ from two distributions and separately. After solving the optimal coupling $:= \min_{\in (\ ;\ )} \int \|x - x'\|_2^2 \, d(x; x')$, the barycentric projection can be expressed as $T_B(X) = N \ X$: We extend the method to two datasets $Z_Q = \{X_Q; Y_Q\}; Z_P = \{X_P; Y_P\}$, where we have additional i.i.d. label data $Y_Q = (y_Q^{(1)}; \dots; y_Q^{(N_Q)}); Y_P = (y_P^{(1)}; \dots; y_P^{(N_P)})$. We first solve the optimal coupling for OTDD (2) following the regularized scheme in Alvarez-Melis & Fusi (2020), and represent labels as one-hot vectors $y \in R^C$. The barycentric projection is divided into two parts:

$$T_B(Z_Q) = [N_Q \ X_P; N_Q \ Y_P]. \qquad (3)$$

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).

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); Gazdieva et al. (2022) approach the Monge OT problem with general cost functions by solving its max-min dual problem $\sup_f \inf_T \int [c(x, T(x)) - f(T(x))]\, d\mu(x) + \int f(x')\, d\nu(x')$. 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 $T_N : \mathbb{R}^d \times \mathbb{R}^{C_Q} \to \mathbb{R}^d \times \mathbb{R}^{C_P}$ as

$$T_N(z) = T_N(x, y) = [x, \ell] = [G(z), \ell(G(z))],$$

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

$$\sup_f \inf_G \int \|x - G(z)\|_2^2 + W_2^2(\mu_y, \nu_\ell)\, dQ(z) - \int f(x, y)\, dQ(z) + \int f(x', y')\, dP(z'). \quad (4)$$

Implementation details are provided in Appendix D. 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 no one-to-one class alignments, e.g., samples can be mapped across similar classes.

3.2 Convex combination in dataset space

Computing generalized geodesics requires constructing convex combinations of data points from different datasets. Given a weight vector $a \in \mathbb{R}^m$, 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 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 $y_1 \in \mathbb{R}^2$ for the first one, we embed it into $\mathbb{R}^9$ as $\tilde{y}_1 = [y_1; 0_3; 0_4]^\top$. 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 loss-less and preserves the distinction of labels across datasets.

3.3 Projection onto generalized geodesic of datasets

We finally put together the components in Sec 3.1 and 3.2 to construct generalized geodesics between datasets in two steps. First, we compute OTDD maps between Q and all other datasets $P_i, i = 1, \ldots, m$ using the discrete or neural OT approaches. Then, for any interpolation vector $a \in \Delta_m$ we identify a dataset along the generalized geodesic via $P_a := (\sum_{i=1}^m a_i T_i)\# Q$. By using the convex combination method in Sec. 3.2 for labeled data, we can efficiently sample from P_a.

We next consider locating the dataset P_a that minimizes the distance between P_a and Q, i.e. the projection of Q onto the generalized geodesic. We firstly 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 $W_2^2(\mu_a, \nu)$. However, it is not straightforward to locate a because there is no closed-form formula of the map $a \mapsto W_2^2(\mu_a, \nu)$ and it can be expensive to calculate $W_2^2(\mu_a, \nu)$ for all possible a. To solve this problem, we resort to another transport distance: $(2, \beta)$-transport metric.

Definition 1 (Craig (2016)). The $(2, \beta)$-transport metric is given by $W_{2,\beta}(\mu_i, \mu_j) := \int \|T_i(x) - T_j(x)\|_2^\beta d\mu(x)^{1/\beta}$, where T_i is the optimal map from μ to μ_i.

When μ has density with respect to Lebesgue measure, $W_{2,\beta}$ is a valid metric (Craig, 2016, Prop. 1.15). Moreover, we can derive the closed-form formula of the map $a \mapsto W_{2,\beta}^\beta(\mu_a, \nu)$.

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

This equation implies that given distributions $\{\mu_i\}_{i=1}^m, \nu$ in Euclidean space, we can trivially solve the optimal a that minimizes $W_{2,\beta}^\beta(\mu_a, \nu)$ by a quadratic programming solver[2]. The proof (Appendix C) relies on Brenier's theorem. Inspired by this, we also define a transport metric for datasets:

[2] We use the implementation in https://github.com/stephane-caron/qpsolvers

(a) (b)

Figure 1: Comparison to mixup interpolation, (a) Approximated projection $\hat{P}_a$ of 3D dataset Q onto the generalized geodesic of datasets $\{P_i\}$. (b) Left to right: the 2D projection of the datasets $(\sum_{i=1}^m a_i T_i)\sharp Q$, $(\sum_{i=1}^m a_i T_i)\sharp Q$, where T_i is the (optimal) OTDD map and $\bar{T}_i$ uses mixup.

Definition 2. The squared $(2,Q)$-dataset distance is given by $W_{2;Q}^2(P_i, P_j) := \int \|x_i - x_j\|_2^2 + W_2^2(\mathcal{Y}_{y_i}, \mathcal{Y}_{y_j})\, dQ(z)$, where $[x_i, y_i] = T_i(z)$ and 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 $d_{OT}(P, Q) < \infty$ 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 also left to the Appendix C.

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

Unfortunately, in this case $W_{2;Q}^2(P_i, P_j)$ does not have an analytic form like before because Brenier's theorem may not hold for a general transport cost problem. However, we still borrow this idea and define an approximated projection $\hat{P}_a$ as the minimizer of function

$$W^2(P_a, Q) := \sum_{i=1}^m a_i W_{2;Q}^2(P_i, Q) - \frac{1}{2} \sum_{i \neq j} a_i a_j W_{2;Q}^2(P_i, P_j); \tag{5}$$

which is an analog of Proposition 1. Unlike the Wasserstein distance, $W_{2;Q}^2(\cdot, \cdot)$ is easier to compute because it does not involve optimization, so it is relatively cheap the locate the minimizer of $W^2(P_a, Q)$. Experimentally, we observe that $W_{2;Q}^2(P_a, Q)$ is predictive of model transferability across tasks. Figure 1(a) illustrates this projection on toy 3D datasets, color-coded by class.

4 Experiments

4.1 Learning OTDD maps on synthetic datasets

Figure 1(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 T_i estimated via barycentric projection (8) results in a better preservation of class structure, whereas using non-optimal maps $\bar{T}_i$ based on random couplings (as the usual mixup does) destroys class structure.

4.2 Transfer learning

Next, we use our framework to generate new pretraining datasets for few-shot learning. Given labeled pretraining datasets $\{P_i\}$, we consider a few-shot test dataset, in which only partial data is labelled, e.g. 5 samples per class. Suppose the training resource and time are both limited such that the user can choose only one dataset to train the model, in the mean time, the user expects the model to have the best generalization ability. To this end, we assume the training dataset is chosen from the generalized geodesics $\{P_a\}$. With a choice of the one-hot weight vector, P_a recovers the original dataset P_i for some i. Otherwise, P_a will be the interpolation of datasets $\{P_i\}$. We first show that the generalization ability of training models has a strong correlation with the distance $W_{2;Q}^2(P_a, Q)$. Then we compare our framework with several baseline methods.

Connection to generalization. The closed-form expression of $W_{2;\mathcal{G}_a}^2(\mathcal{G}_a, \cdot)$ (Prop. 1) provides the distance between a base distribution and the distribution along generalized geodesic in Euclidean space. We study its analog (5) for labelled datasets Q and $\{P_i\}$ in Figure 2. To investigate

Figure 2: Relationship between the function $W_2^2(P_a, Q)$ and the accuracy of the fine-tuned model. The training datasets are marked on the vertexes of each ternary plot. Different location in each ternary plot represents different interpolation dataset P_a, where the center is the most mixed dataset, i.e. $a = [1/3, 1/3, 1/3]$. We visualize the function (5) in the first row and the fine-tuning accuracy in the second. Comparing the first row and the second, we find the accuracy and $W_2^2(P_a, Q)$ are highly correlated. This implies that the model trained on the minimizer dataset of $W_2^2(P_a, Q)$ tends to have a better generalization ability. Each ternary plot is an average of 5 runs with distinct random seeds.

Table 1: Pretraining on synthetic data. Shown is 5-shot transfer accuracy (mean ± s.d. over 5 runs).

Methods	MNIST-M	MNIST	USPS	FMNIST	KMNIST	EMNIST
OTDD barycentric projection	42.10 ± 4.37	93.74 ± 1.46	86.01 ± 1.50	70.12 ± 3.02	52.55 ± 2.73	67.06 ± 2.55
OTDD neural map	40.06 ± 4.75	88.78 ± 3.85	83.80 ± 1.60	70.02 ± 2.59	50.32 ± 3.10	65.32 ± 1.80
Mixup	33.85 ± 2.22	88.68 ± 1.57	88.61 ± 2.00	66.74 ± 3.79	48.16 ± 3.38	60.95 ± 1.38
Train on few-shot dataset	19.13 ± 3.57	72.80 ± 3.10	80.73 ± 2.07	60.50 ± 3.07	41.67 ± 2.11	53.60 ± 1.18
1-NN on few-shot dataset	20.95 ± 1.39	64.50 ± 3.32	73.64 ± 2.35	60.92 ± 2.42	40.18 ± 3.09	39.70 ± 0.57

the generalization abilities of models trained on different datasets, we discretize the simplex Δ_3 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. We fix the same colorbar range for all heatmaps across datasets to highlight the different impact of choosing training dataset. For some test datasets, the choice of training dataset can affect the fine-tuning accuracy greatly. For example, when Q is EMNIST and the training dataset is FMNIST, the fine-tuning accuracy is only 60%, but this can be improved to 70% by choosing an interpolated dataset closer to MNIST. This is reasonable because MNIST shares more similarity with 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 experiment, we select one dataset as the test dataset, and the rest NIST datasets are the training datasets. We assume the test dataset is 5-shot, and to do this, 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 P_a, and fine-tune the model on 5-shot test dataset. To obtain P_a, we use barycentric projection or neural map to approximate the OTDD maps from test dataset to the training datasets. Our results are shown in the first two rows in Table 1. The first baseline method is to create a synthetic dataset as training dataset by Mixup among datasets. We randomly sample data from each training datasets, and do the convex combination of them with weight a. We use the same convex combination method in Sec. 3.2, thus this baseline is equivalent to our framework with suboptimal OTDD maps. The other two baselines (the bottom block in Table 1) skip the transfer learning part, and directly train the model or solve 1-NN on the few-shot test dataset. 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%

References

Agueh, M. and Carlier, G. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011. (Cited on pages 2, 10, and 11.)

Alvarez-Melis, D. and Fusi, N. Geometric dataset distances via optimal transport. *Advances in Neural Information Processing Systems*, 33:21428–21439, 2020. (Cited on pages 1, 2, and 3.)

Alvarez-Melis, D. and Fusi, N. Dataset dynamics via gradient flows in probability space. In *International Conference on Machine Learning*, pp. 219–230. PMLR, 2021. (Cited on page 10.)

Ambrosio, L., Gigli, N., and Savaré, G. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008. (Cited on pages 2, 3, and 10.)

Asadulaev, A., Korotin, A., Egiazarian, V., and Burnaev, E. Neural optimal transport with general cost functionals. *arXiv preprint arXiv:2205.15403*, 2022. (Cited on pages 4, 10, and 13.)

Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R., Hammers, A., Dickie, D. A., Hernández, M. V., Wardlaw, J., and Rueckert, D. GAN augmentation: Augmenting training data using generative adversarial networks, October 2018. (Cited on page 10.)

Brenier, Y. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417, 1991. (Cited on page 2.)

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are Few-Shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. (Cited on page 1.)

Bunne, C., Krause, A., and Cuturi, M. Supervised training of conditional monge maps. *arXiv preprint arXiv:2206.14262*, 2022a. (Cited on pages 4, 10, and 13.)

Bunne, C., Papaxanthos, L., Krause, A., and Cuturi, M. Proximal optimal transport modeling of population dynamics. In Camps-Valls, G., Ruiz, F. J. R., and Valera, I. (eds.), *Proceedings of The 25th International Conference on Arti cial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 6511–6528. PMLR, 2022b. (Cited on page 10.)

Chuang, C.-Y. and Mroueh, Y. Fair mixup: Fairness via interpolation. In *International Conference on Learning Representations*, 2021. (Cited on page 10.)

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pp. 2921–2926. IEEE, 2017. (Cited on page 12.)

Craig, K. The exponential formula for the wasserstein metric. *ESAIM: Control, Optimisation and Calculus of Variations*, 22(1):169–187, 2016. (Cited on pages 2, 4, and 10.)

Dvurechensky, P., Gasnikov, A., and Kroshnin, A. Computational optimal transport: Complexity by accelerated gradient descent is better than by sinkhorn's algorithm. In *International conference on machine learning*, pp. 1367–1376. PMLR, 2018. (Cited on page 10.)

Fan, J., Taghvaei, A., and Chen, Y. Scalable computations of wasserstein barycenter via input convex neural networks. *arXiv preprint arXiv:2007.04462*, 2020. (Cited on pages 2 and 3.)

Fan, J., Liu, S., Ma, S., Chen, Y., and Zhou, H. Scalable computation of monge maps with general costs. *arXiv preprint arXiv:2106.03812*, 2021. (Cited on pages 4 and 10.)

Fan, J., Zhang, Q., Taghvaei, A., and Chen, Y. Variational Wasserstein gradient flow. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 6185–6215. PMLR, 2022. (Cited on page 10.)

Gao, Y. and Chaudhari, P. An Information-Geometric distance on the space of tasks. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021. (Cited on page 1.)

Gazdieva, M., Rout, L., Korotin, A., Filippov, A., and Burnaev, E. Unpaired image super-resolution with optimal transport maps. *arXiv preprint arXiv:2202.01116*, 2022. (Cited on page 4.)

Hull, J. J. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994. (Cited on page 12.)

Jain, S., Salman, H., Khaddaj, A., Wong, E., Park, S. M., and Madry, A. A data-based perspective on transfer learning. *arXiv preprint arXiv:2207.05739*, 2022. (Cited on page 1.)

Kabir, H. D., Abdar, M., Khosravi, A., Jalali, S. M. J., Atiya, A. F., Nahavandi, S., and Srinivasan, D. Spinalnet: Deep neural network with gradual input. *IEEE Transactions on Arti cial Intelligence*, 2022. (Cited on page 12.)

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *PNAS*, 114(13):3521–3526, mar 2017. (Cited on page 1.)

Korotin, A., Li, L., Solomon, J., and Burnaev, E. Continuous wasserstein-2 barycenter estimation without minimax optimization. *arXiv preprint arXiv:2102.01752*, 2021. (Cited on page 3.)

Korotin, A., Egiazarian, V., Li, L., and Burnaev, E. Wasserstein iterative networks for barycenter estimation. *arXiv preprint arXiv:2201.12245*, 2022a. (Cited on page 3.)

Korotin, A., Selikhanovych, D., and Burnaev, E. Neural optimal transport. *arXiv preprint arXiv:2201.12220*, 2022b. (Cited on pages 2 and 10.)

Liu, H., Gu, X., and Samaras, D. Wasserstein gan with quadratic transport cost. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4832–4841, 2019. (Cited on page 12.)

Makkuva, A., Taghvaei, A., Oh, S., and Lee, J. Optimal transport mapping via input convex neural networks. In *International Conference on Machine Learning*, volume 37, 2020. (Cited on pages 2 and 10.)

McCann, R. J. Existence and uniqueness of monotone measure-preserving maps. *Duke Mathematical Journal*, 80(2):309–323, 1995. (Cited on page 11.)

McCann, R. J. A convexity principle for interacting gases. *Advances in mathematics*, 128(1):153–179, 1997. (Cited on pages 2 and 10.)

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. In Bower, G. H. (ed.), *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Academic Press, January 1989. doi: 10.1016/S0079-7421(08)60536-8. (Cited on page 1.)

Mokrov, P., Korotin, A., Li, L., Genevay, A., Solomon, J. M., and Burnaev, E. Large-Scale wasserstein gradient flows. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P. S., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 15243–15256. Curran Associates, Inc., 2021. (Cited on page 10.)

Perrot, M., Courty, N., Flamary, R., and Habrard, A. Mapping estimation for discrete optimal transport. *Advances in Neural Information Processing Systems*, 29, 2016. (Cited on page 10.)

Pooladian, A.-A. and Niles-Weed, J. Entropic estimation of optimal transport maps. *arXiv preprint arXiv:2109.12004*, 2021. (Cited on page 3.)

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015. (Cited on page 12.)

Rout, L., Korotin, A., and Burnaev, E. Generative modeling with optimal transport maps. In International Conference on Learning Representations, 2022. (Cited on pages 4 and 10.)

Sandfort, V., Yan, K., Pickhardt, P. J., and Summers, R. M. Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. Sci. Rep., 9(1):16884, November 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-52737-x. (Cited on page 10.)

Santambrogio, F. Optimal transport for applied mathematicians. Birkäuser, NY, 55(58-63):94, 2015. (Cited on pages 2 and 10.)

Santambrogio, F. Euclidean, metric, and Wasserstein gradient flows: an overview. Bulletin of Mathematical Sciences, 7(1):87–154, 2017. (Cited on pages 3 and 11.)

Sinkhorn, R. Diagonal equivalence to matrices with prescribed row and column sums. The American Mathematical Monthly, 74(4):402–405, 1967. (Cited on page 3.)

Srivastava, S., Li, C., and Dunson, D. B. Scalable bayes via barycenter in wasserstein space. The Journal of Machine Learning Research, 19(1):312–346, 2018. (Cited on page 3.)

Villani, C. Optimal transport, Old and New, volume 338. Springer Science & Business Media, 2008. ISBN 9783540710493. (Cited on pages 2 and 10.)

Yoon, J., Jordon, J., and van der Schaar, M. PATE-GAN: Generating synthetic data with differential privacy guarantees. In International Conference on Learning Representations, 2019. (Cited on page 10.)

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In International Conference on Learning Representations, 2018. (Cited on pages 10 and 14.)

Zhang, L., Deng, Z., Kawaguchi, K., Ghorbani, A., and Zou, J. How does mixup help with robustness and generalization? In International Conference on Learning Representations, 2021. (Cited on page 10.)

A 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 & Mroueh, 2021) use mixup to improve in-domain model robustness and generalization by increasing 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 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 con ned to generate 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 typical effective method to approximate optimal transport map with discrete regularized OT. Other than this, neural net based optimal map in Euclidean space has made great progress (Makkuva et al., 2020; Fan et al., 2021; Rout et al., 2022) recently and reveal its power in image generation (Rout et al., 2022), style transfer (Korotin et al., 2022b), etc. However, the study of the optimal map between two datasets is relatively scarse. Some conditional Monge map solvers (Asadulaev et al., 2022; 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 dataset mapping is distinct from them because we do not enforce the label to label mapping. Based on the optimal coupling or map, geodesics and interpolation in general metric spaces have been studied extensively in the optimal transport and metric geometry literatures (McCann, 1997; Agueh & Carlier, 2011; Ambrosio et al., 2008; Santambrogio, 2015; Villani, 2008; Craig, 2016), albeit mostly in a theoretical setting. Gradient ows (Santambrogio, 2015), as an alternative approach for interpolation between distributions, have become increasingly popular in machine learning to model existing processes (Bunne et al., 2022b; Mokrov et al., 2021; Fan et al., 2022) or solving optimization problems over datasets (Alvarez-Melis & Fusi, 2021), but they are computationally heavy.

B Discussion

Complexity The complexity of solving OTDD barycentric projection by Sinkhorn algorithm is $O(N^2)$ (Dvurechensky et al., 2018), where N is the number of data in both datasets. This can be expensive for large-scale dataset. In practice, we solve the batched barycentric projection, i.e. take a batch from source and target datasets and solve the projection from source batch to target batch, and we normally x batch size B as 10^4. This reduces the complexity from $O(N^2)$ to $O(BN)$. The complexity of solving OTDD neural map is $O(BKH)$, where K is number of iterations, and H is the size of the network. We always choose $K = O(N)$ in the experiments. The complexity of solving all the $(2; Q)$-dataset distances in (5) is $O(m^2N)$ 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 θ for the minimizer of (5) is $O(N(B + m^2))$.

Limitation The generation of synthetic dataset relies on solving OTDD maps from 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 ne-tuning pipeline. This can be resource demanding for large-scale models, like GPT model.

C Proofs

Proof of Lemma 1. By Santambrogio (2017, Sec. 4.4), the result holds when $m = 2$. Then Proposition 7.5 in Agueh & Carlier (2011) extends the result to the case of $m > 2$. □

Proof of Proposition 1. Since linear combination preserves cyclically monotonicity, $\sum_{i=1}^{m} a_i T_i^*(x)$ is the optimal map from to $\frac{G}{a}$ (McCann, 1995). Then according to the definition of $W_{2;}(\ ;\ )$, we can write

$$W_{2;}^2(\ \tfrac{G}{a};\ ) = \int x \quad \sum_{i=1} a_i T_i^*(x) \quad ^2 \mathrm{d}\ (x): \tag{6}$$

For scalars $p; q_1; : : : ; q_m$, it holds that

$$\begin{aligned} p \quad \sum_{i=1} a_i q_i \quad ^{!2} &= p^2 + \sum_{i=1} a_i^2 q_i^2 \quad 2 \sum_{i=1} a_i p q_i + \sum_{i \neq j} a_i a_j q_i q_j \\ &= p^2 + \sum_{i=1} (a_i \quad a_i \quad \sum_{j \neq i} a_j) q_i^2 \quad 2 \sum_{i=1} a_i p q_i + \sum_{i \neq j} a_i a_j q_i q_j \\ &= \sum_{i=1} a_i (p \quad q_i)^2 \quad \frac{1}{2} \sum_{i \neq j} a_i a_j (q_i \quad q_j)^2: \end{aligned}$$

Plugging this equality into (6) gives

$$\begin{aligned} W_{2;}^2(\ \tfrac{G}{a};\ ) &= \int \Big(\sum_{i=1} a_i k x \quad T_i^*(x) k^2 \quad \frac{1}{2} \sum_{i \neq j} a_i a_j k T_i^*(x) \quad T_j^*(x) k^2 \Big) \mathrm{d}\ (x) \\ &= \sum_{i=1} a_i \quad \int k x \quad T_i^*(x) k^2 \mathrm{d}\ (x) \quad \frac{1}{2} \sum_{i \neq j} a_i a_j \quad \int k T_i^*(x) \quad T_j^*(x) k^2 \mathrm{d}\ (x) \\ &= \sum_{i=1} a_i W_{2;}^2(\ _i;\ ) \quad \frac{1}{2} \sum_{i \neq j} a_i a_j W_{2;}^2(\ _i;\ _j): \end{aligned}$$

□

Proof of Proposition 2. Firstly, $W_{2;Q}$ is symmetric and nonnegative by definition. It is non-degenerate since $W_{2;Q}(P_i; P_j) \quad d_{\mathrm{OT}}(P_i; P_j)$ and d_{OT} is a metric. Finally, we show it satisfies the triangular inequality. Indeed,

$$\begin{aligned} & W_{2;Q}(P_1; P_3) \\ = & \int kx_1 \quad x_3 k^2 + W_2^2(\ _{y_1};\ _{y_3}) \mathrm{d}Q(z) \quad ^{1=2} \\ & \int (kx_1 \quad x_2 k + kx_2 \quad x_3 k)^2 + (W_2(\ _{y_1};\ _{y_2}) + W_2(\ _{y_2};\ _{y_3}))^2 \mathrm{d}Q(z) \quad ^{1=2} \\ & \int kx_1 \quad x_2 k^2 + W_2^2(\ _{y_1};\ _{y_2}) \mathrm{d}Q(z) \quad ^{1=2} + \int kx_2 \quad x_3 k^2 + W_2^2(\ _{y_2};\ _{y_3}) \mathrm{d}Q(z) \quad ^{1=2} \\ = & W_{2;Q}(P_1; P_2) + W_{2;Q}(P_2; P_3); \end{aligned}$$

where the first inequality is the triangular inequality and the second inequality is the Minkowski inequality. □

D Implementation details of OTDD map

OTDD barycentric projection We use the implementation https://github.com/microsoft/otdd to solve OTDD coupling. The rest part is straightforward.

OTDD neural map To solve the problem (4), we parameterize $f; G; \grave{}$ to be three neural networks. In NIST dataset experiments, we parameterize f as ResNet [3] from WGAN-QC (Liu et al., 2019), and take feature map G to be UNet[4] (Ronneberger et al., 2015). We generate the labels y with a pre-trained classifier $\grave{}(\ )$, and use a LeNet or VGG-5 with Spinal layers[5] (Kabir et al., 2022) to parameterize $\grave{}(\ )$. In 2D Gaussian mixture experiments, we use Residual MLP to represent all of them.

We remove the discriminator's condition on label to simplify the loss function as

$$\sup_f \inf_G \int \underbrace{\|x - G(z)\|_2^2}_{\text{feature loss}} + \underbrace{W_2^2(\ y;\ y)}_{\text{label loss}}\, \mathrm{d}Q(z) \underbrace{\int f(x)\mathrm{d}Q(z) + \int f(x')\mathrm{d}P(z')}_{\text{discriminator loss}} :$$

In this formula, we assume both y and y are hard labels, but in practice, the output of $\grave{}(\ )$ is a soft label. Simply taking the `argmax` to get a hard label can break the computational graph, so we replace the label loss $W_2^2(\ y;\ y)$ by $y^\top M y$, where $M \ 2\ \mathrm{R}^{C_Q \times C_P}$ is the label-to-label matrix where $M(i; j) := W_2^2(\ y_i;\ y_j)$; and y is the one-hot label from dataset Q. The matrix M is precomputed before the training, and is frozen during the training.

We pre-train the feature map G to be identity map before the main adversarial training. We use the Exponential Moving Average[6] of the trained feature maps as the final feature map.

Data processing For all the NIST datasets, we rescale the images to size 32 32, and repeat their channel 3 times and obtain 3-channel images. We use the default train-test split from `torchvision`.

Hyperparameters For the experimental results in Sec. 4.2, we use the OTDD neural map and train them with learning rate 10^{-3} and batch size 64. We train a LeNet for 2000 iterations, and fine-tune for 100 epochs. Regard the comparison with other baselines in Sec. 4.2, for transfer learning methods, we train a SpinalNet for 10^4 iterations, and fine-tune it for 2000 iterations on test dataset. Training from scratch on the test dataset takes also 2000 iterations.

E Additional results

E.1 OTDD neural map visualization

In Figure 4, we in addition 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 Sec. 3.1. 1) We don't assume a known source label to target label correspondence. So we can map between two irrelevent datasets such as EMNIST and FashinMNIST. 2) The map is invariant to the permutation of label assignment. For example, we show two different labelling in Figure 3, and the final OTDD map will be the same. 3) It doesn't enforce the label to label mapping but would follow the feature similarity. From Figure 4 in the appendix, 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.

Figure 3: The numbers above images are the labels. In the first labelling method, all 0 MNIST digits are assigned as class "0", and they are labelled as class "7" in the bottom labelling.

We also show the OTDD neural map between 2D Gaussian mixture models with 16 components in Figure 5. This example is very special so that we have the closed-form solution of OTDD map. The feature map is a identity map and the pushforward label is equal to the corresponding class that has

[3] https://github.com/harryliew/WGAN-QC
[4] https://github.com/milesial/Pytorch-UNet
[5] https://github.com/dipuk0506/SpinalNet
[6] https://github.com/fadel/pytorch_ema

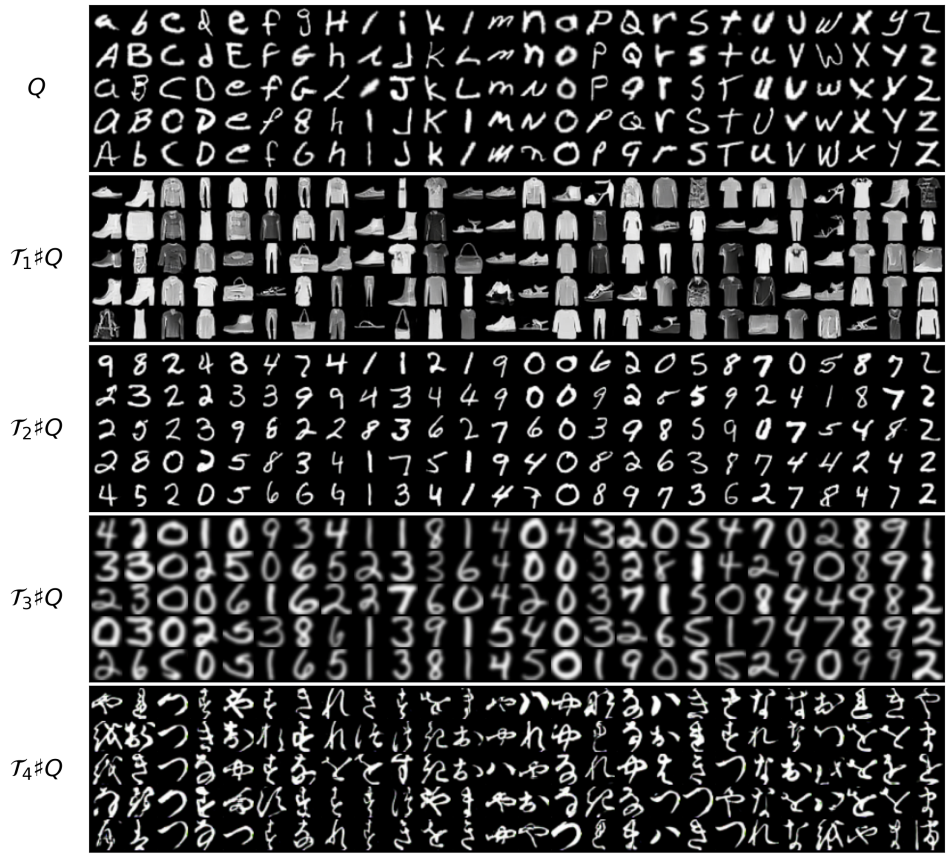

Figure 4: The dataset Q is EMNIST (letters). We show all the datasets pushforwarded towards Fashion-MNIST, MNIST, USPS, KMNIST by OTDD map. The OTDD map is solved by neural OT method.

the same conditional distribution $p(xjy)$ as source label. For example, the sample from top left corner cluster is still mapped to the top left corner cluster, and the label is changed from blue to orange. This map achieves zero transport cost. Since the transport cost is always non-negative, this map is the optimal OTDD map. However, Asadulaev et al. (2022); Bunne et al. (2022a) enforce mapping to preserve the labels, so with their methods, the blue cluster would still map to the blue cluster. Thus their feature map is highly non-convex and more difficult to learn. We refer to Figure 5 in Asadulaev et al. (2022) for their performance on the same example. Compared with them, our pushforward dataset aligns with the target dataset better.

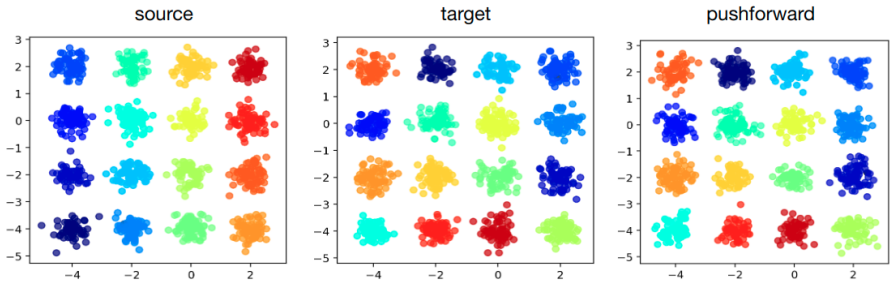

Figure 5: OTDD neural map for 2D Gaussian mixture distributions.