# Optimal Transport under Data Locality

**Chuan Lin, Rishi More**
clin146, rmore2

## Abstract

Optimal transport is a widely used method for domain adaptation, mapping source data points to target data points while minimizing the transport cost. However, many approaches fail to preserve data locality while transporting the mass of source points to the mass of target points. In other words, current optimal transport algorithm may lead to a transport that even two source points that are really close in source domain may transport their own mass to target points extremely differently. In this proposal, we introduce a cluster-based algorithm for optimal transport that not only ensures data locality but also shares all kinds of advantages, including low computational cost and compatibility with any existing optimal transport algorithm.

## 1 Introduction

**Optimal Transport** Optimal transport is a widely used method for domain adaptation. Specifically, given a labeled training dataset $X_s = \{x_i^s\}_{i=1}^{N_s}, \quad Y_s = \{y_i^s\}_{i=1}^{N_s}$ and an unlabeled test dataset $X_t = \{x_i^t\}_{i=1}^{N_t}$ where information preservation property holds (That is, there exists a transportation map $T : \Omega_s \to \Omega_t$ from the source space $\Omega_s$ to the target space $\Omega_t$, such that the conditional distribution of labels remains unchanged after mapping $P_s(y \mid x^s) = P_t(y \mid T(x^s))$), the optimal transport method provides a principled approach to domain adaptation, enabling the construction of a classifier that generalizes well to the target domain. The procedure consists of the following steps. An illustration of the workflow of optimal transport is shown in Figure 1.

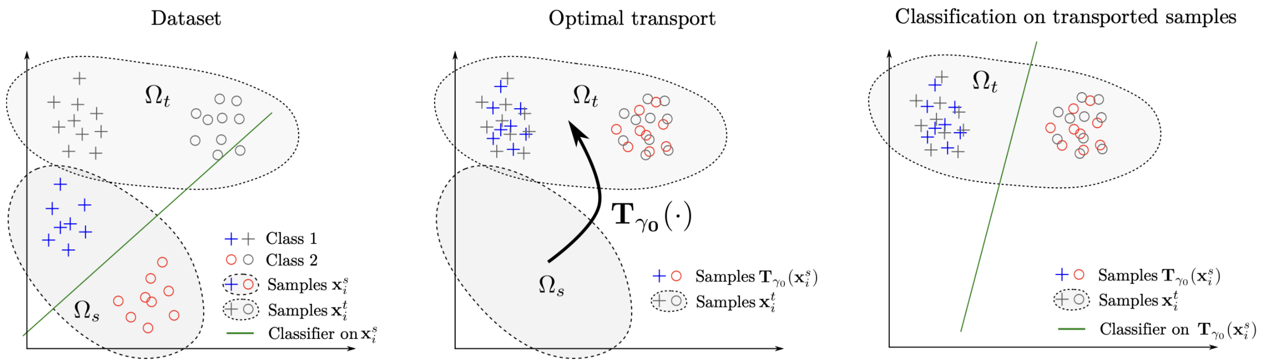

Dataset
$\Omega_t$
$\Omega_s$
Class 1
Class 2
Samples $\mathbf{x}_i^s$
Samples $\mathbf{x}_i^t$
Classifier on $\mathbf{x}_i^s$
Optimal transport
$\Omega_t$
$\mathbf{T}_{\gamma_0}(\cdot)$
$\Omega_s$
Samples $\mathbf{T}_{\gamma_0}(\mathbf{x}_i^s)$
Samples $\mathbf{x}_i^t$
Classification on transported samples
$\Omega_t$
Samples $\mathbf{T}_{\gamma_0}(\mathbf{x}_i^s)$
Samples $\mathbf{x}_i^t$
Classifier on $\mathbf{T}_{\gamma_0}(\mathbf{x}_i^s)$

Illustration of the proposed approach for domain adaptation. (left) dataset for training, *i.e.* source domain, and testing, *i.e.* target domain. Note that a classifier estimated on the training examples clearly does not fit the target data. (middle) a data dependent transportation map $\mathbf{T}_{\gamma 0}$ is estimated and used to transport the training samples onto the target domain. Note that this transformation is usually not linear. (right) the transported labeled samples are used for estimating a classifier in the target domain.

Figure 1: Workflow for Optimal Transport

1. Estimate the marginal distributions of the training and test data separately:
$$\mu_s(x^s) = P_s(x^s), \quad \mu_t(x^t) = P_t(x^t)$$
2. Find an optimal transport map $T$, which aligns data points in the source space with those in the target space:
$$x^s \in \Omega_s, \quad T(x^s) \in \Omega_t$$
3. Transform the labeled training dataset $(X_s, Y_s)$ using $T$ to obtain a new labeled dataset $(X_s^{'}, Y_s^{'})$:
$$X_s' = T(X_s), \quad Y_s' = Y_s$$

   Using this transformed dataset, a classifier is then trained to generalize effectively to the target domain.

The key step in this principled process is Step 2, determining the optimal transport map. To achieve this, we introduce the following two steps.

1. Firstly, we introduce the Kantorovich formulation of the optimal transport problem. Under this formulation, finding the optimal transport map $T$ reduces to determining the optimal transport matrix $\gamma$, where matrix $\gamma$ belongs to the set $\beta$ of probabilistic couplings of empirical distributions:

$$\beta = \{\gamma \in (R^{+})^{N_s \times N_t} \mid \gamma \mathbf{1} = \mu_s, \quad \gamma^T \mathbf{1} = \mu_g\}$$

$$\mu_s \in \Delta^{N_s}, \quad \mu_t \in \Delta^{N_t}$$

2. To select the optimal transport matrix, we introduce the transport cost as the key criterion. With this criterion, the optimal transport matrix $\gamma_0$ is given by:

$$\gamma_0 = argmin_{\gamma \in \beta} \langle \gamma, C \rangle_F,$$

where $C$ is the cost matrix, defined as $C(i,j) = c(x_i^s, x_j^t)$, which represents the cost of transporting unit probability mass from point $x_i^s$ in the source space $\Omega_s$ to point $x_j^t$ in the target space $\Omega_t$. The cost function $c(\cdot,\cdot)$ is pre-defined, often leveraging prior knowledge. When the source and target spaces coincide, i.e. $\Omega_s = \Omega_t$, the Euclidean distance is commonly used as the cost function.

However, the transport cost is not the only criterion about whether a transport map is good or not. For instance, regularized optimal transport suggests that beyond minimizing the transport cost, we must also ensure the smoothness of the transport matrix. Consequently, the optimal transport matrix takes the following form, where $\Omega(\gamma)$ is a regularization term like the negative entropy: $\gamma_0 = argmin_{\gamma \in \beta} \langle \gamma, C \rangle_F + \lambda \Omega(\gamma)$. Besides, self-supervised optimal transport suggests that samples in the target domain should only be matched with samples in the source domain that have the same labels. In this case, $\gamma_0 = argmin_{\gamma \in \beta} \langle \gamma, C \rangle_F + \langle \gamma, M \rangle$, where $M$ is a $n_s \times n_t$ matrix ($M_{ij} = 0$ when $y_i^s = y_j^t$). Causal transport suggests that if there is a causal relationship between different dimensions of feature space, the transport should also be causal. In other words, when we view the transport as a joint distribution of source points and target points, some kind of conditional independence that must be satisfied.

Although the existing work has already explored all kinds of alternative criteria for choosing optimal map, none of them incorporates data locality. In other words, current optimal transport algorithm may lead to a transport that even two source points that are really close in source domain may transport their own mass to target points totally differently.

**Data Locality** Naturally speaking, data locality means that data points close in feature space tend to be similar in label. Mathematically speaking, data locality means the continuity of the function describing the relationship between features and label: $\lim_{x \to x_0} F(x) = F(x_0)$. When we are dealing with regression problem, function $F$ is a mapping from feature space to label space; when we are dealing with classification problem, function $F$ is a mapping from feature space to $K-1$-dimensional simplex, where $K$ is the number of classes.

Given a dataset, a good way to see whether there is a strong data locality is to draw a scatter plot, where each point of the plot corresponds with a pair of data points, whose x-value means the similarity of these two points in feature space and y-value means the similarity of these two points in label space. As a result, we can tell whether there is a positive correlation by telling it directly or calculating the correlation coefficient.

Many algorithms or concepts utilized data locality. For example, K nearest neighboring algorithm (KNN) tells the label of the test point by looking at the labels of the neighbors of this point in the training dataset. Another example is the concept of aleatoric uncertainty and epistemic uncertainty. Epistemic uncertainty means that when we are trying to tell the label of a test point in whose neighborhood we don't have enough training points, we should be less confident with our prediction.

**Our Contribution** Our contribution can be summarized into the following three aspects.

1. We propose a cluster-based transport algorithm that not only firstly ensures data locality during transports but also shares all kinds of merits such as low computational cost and compatibility with any existing optimal transport algorithm.
2. We propose information loss as a new criterion for selecting the transport. This new loss can only be applied to our cluster-based transport algorithm. Like self-supervised regularization, our information loss term ensures information preservation assumption holds. In other words, it ensures our learned transportation map $T : \Omega_s \to \Omega_t$ from the source space $\Omega_s$ to the target space $\Omega_t$ is one such that the conditional distribution of labels remains unchanged after mapping $P_s(y \mid x^s) = P_t(y \mid T(x^s))$. However, the advantage of our information loss to self-supervised regularization is that we don't need the labels of test data points.
3. We do all kinds of experiments to show the superiority of our algorithm.

## 2 Related Work

Current work has already defined the optimal transport problem very well and explored all kinds of its variants [3]. Most work agree with the transport cost as a criterion for choosing the transport. However, there are all kinds of variants. For instance, regularized optimal transport suggests that beyond minimizing the transport cost, we must also ensure the smoothness of the transport matrix [3]. Consequently, the optimal transport matrix takes the form, where $\Omega(\gamma)$ is a regularization term like the negative entropy: $\gamma_0 = argmin_{\gamma \in \beta} \langle \gamma, C \rangle_F + \lambda \Omega(\gamma)$.

Besides, self-supervised optimal transport suggest that samples in the target domain should only be matched with samples in the source domain that have the same labels [3]. In this case, $\gamma_0 = argmin_{\gamma \in \beta}\langle \gamma, C\rangle_F + \langle \gamma, M\rangle$, where $M$ is a $n_s \times n_t$ matrix ($M_{ij} = 0$ when $y_i^s = y_j^t$). Causal transport suggests that if there is a causal relationship between different dimensions of feature space, the transport should also be causal [4]. In other words, when we view the transport as a joint distribution of source points and target points, some kind of conditional independence that must be satisfied.

## 3 Proposed Work

### 3.1 Cluster-Based Algorithm for Data Locality

As mentioned above, optimal transport deals with the situation that we are given a labeled training dataset $X_s = \{x_i^s\}_{i=1}^{N_s}, \quad Y_s = \{y_i^s\}_{i=1}^{N_s}$ and an unlabeled test dataset $X_t = \{x_i^t\}_{i=1}^{N_t}$.

Our cluster-based algorithm is made up of the following two steps.

1. Step 1: Cluster our training data points into $N_c$ clusters, where $N_c$ is a hyper-parameter. In other words, from $X_s = \{x_i^s\}_{i=1}^{N_s}, \quad Y_s = \{y_i^s\}_{i=1}^{N_s}$, we get $X_c = \{x_i^c\}_{i=1}^{N_c}, \quad Y_c = \{y_i^c\}_{i=1}^{N_c}$.
   To be more specific, $x_i^c$ is the centroid of the $i$-th cluster. $y_i^c$ is obtained from all source points $y^s$ in the $i$-th cluster. If we are dealing with a regression problem, $y_i^c$ is simply the average of all the corresponding $y^s$; if we are dealing with a classification problem, $y^s$ can be regarded as a $K$ dimensional one-hot vector ($K$ is the number of classes) and $y_i^c$ is also the average of all corresponding $y^s$, which makes $y_i^c$ a vector representing a categorical distribution over K classes.
2. Step 2: Based on our transformed dataset $X_c = \{x_i^c\}_{i=1}^{N_c}, \quad Y_c = \{y_i^c\}_{i=1}^{N_c} \quad X_t = \{x_i^t\}_{i=1}^{N_t}$ , we can apply any existing algorithm to tell how to transport centroids to target points. For example, if we are using traditional OT algorithm, our optimization problem is $\gamma_0 = argmin_{\gamma \in \beta}\langle \gamma, C\rangle_F$ where both matrix $\gamma$ and matrix $C$ are $N_c \times N_t$.
   After we get the transport plans of all centroids, we let each source point $x^s$ has the same transport plan as their corresponding cluster $x^c$. In other words, assume that $\gamma^1 \in R^{N_s \times N_t}$ is our transport from source points to target points and $\gamma^2 \in R^{N_c \times N_t}$ is our transport from centroids to target points, then the $i$-th row of $\gamma^1$ will be the same of the $j$-th row of $\gamma^2$ as long as the i-th source point is assigned to the j-th cluster.

Our cluster-based algorithm is simple but effective at ensuring data locality. When data points are close, they will be assigned into the same cluster and have the same transport plan, which ensures the data locality. Besides, our simple method possesses the following merits. Firstly, it reduces the computational cost since the number of clusters is much less than the number of source data points. Secondly, it is compatible with any existing OT algorithm because any OT algorithms can tell how to transport centroids to target points by just regarding the centroids as training dataset. Lastly, it can ensures the information preservation assumption without the need of labels of target points, which is achieved with our information loss term in the following section.

### 3.2 Information Loss to Ensure Information Preservation

**Necessary Condition from Information Preservation Assumption** As mentioned above, the information preservation assumption is made as follows:

$P_s(y|x^s) = P_t(y|T(x^s))$

To directly check whether this assumption holds or not, we need to iterate over all possible source data points to compare the conditional label distribution under source data point $P_s(y|x^s)$ and that under the corresponding target point $P_t(y|T(x^s))$, which is the idea of self-supervised regularization. However, if we don't have labels on target space, this direct check is not viable.

However, some necessary conditions are derived from this information preservation assumption. Assume that we have two different source data points $x_1^s, \quad x_2^s$ in the source space that are mapped to the same target data point in the target space, namely $T(x_1^s) = T(x_2^s)$. Then we have the following derivation:

$P_s(y|x_1^s) = P_t(y|T(x_1^s)) = P_t(y|T(x_2^s)) = P_s(y|x_2^s)$

As a result, we can see that under the information preservation assumption, two different source data points that are mapped to the same target data point must have the same source conditional label distribution. When applied to our cluster-based algorithm, we can see that two different clusters that are mapped to the same target data point must have the same within-cluster label distribution.

**Information Loss for Transporting Centroids to Target Points** The above analysis tells us that we need to ensure that two different clusters that are mapped to the same target data point must have the same within-cluster label distribution.

When dealing with the optimal transport problem to transport centroids to target points, namely when dealing with the following transformed data set $X_c = \{x_i^c\}_{i=1}^{N_c}, \quad Y_c = \{y_i^c\}_{i=1}^{N_c} \quad X_t = \{x_i^t\}_{i=1}^{N_t}$, we can define an information loss term for the transportation matrix $\gamma \in R^{N_C \times N_t}$ in the following three steps:

1. Firstly, we define the coupling matrix $M$ as follows, which couples any pair of cluster with a scalar:
   $M = \gamma \times \gamma^T, \quad M \in (R^+)^{N_c \times N_c}$
   In this case, the term $M(i, j)$ indicates how strongly the cluster $x_i^c$ and the cluster $x_j^c$ are mapped to the same target data point. If they are mapped to different target data points, $M(i, j)$ will be 0.

2. Secondly, we define the divergence matrix H as follows:
   $H(i,j) = KL(y_i^c||y_j^c), \quad H \in (R^+)^{N_c \times N_c}$

   In this case, the term $H(i,j)$ indicates how different the label distribution within cluster $i$ and the label distribution within cluster $j$ are. As mentioned above $y_i^c$ is a vector that represents a categorical distribution within cluster $i$ because it is a $K$-dimensional vector where $K$ is the number of classes
3. Lastly, we have our information loss term as follows. The bigger it is, the less likely our information preservation assumption is met.
   $L = \langle M, H \rangle_F, \quad L \in R^+$

With this term, we can find the transport matrix from centroids to target points that not only needs the lowest transport cost but also preserves information as much as possible, as follows:

$\gamma_0 = argmin_{\gamma \in \beta} \langle \gamma, C \rangle_F + \langle M, H \rangle_F$

**Comparision with Self-Supervised Regularization** The self-supervised regularization defines their term by directly applying the definition of information preservation assumption while our information loss term is defined by applying the necessary condition of this assumption. As a result, their method needs the labels of target points to build a $N_s \times N_t$ matrix while our method only needs the labels of source points to build a $N_c \times N_c$ matrix.

## 4 Experimental Results

labelsec:experiments

We evaluate our proposed optimal transport methods on both synthetic and real-world domain adaptation tasks. All evaluations are performed using a 1-nearest neighbor (1-NN) classifier trained on the transported source samples and tested on the target domain. This provides a quantitative measure of domain alignment quality.

We compare the following methods:

- **No Adaptation:** Classifier trained solely on the source domain.
- **Sinkhorn OT:** Standard entropic regularized optimal transport.
- **OT + Info–Loss (Ours):** Optimal transport with our proposed information loss regularization.
- **Cluster OT + Info–Loss (Ours):** OT with both information loss and cluster consistency regularization.

### 4.1 Synthetic Two–Moons

**Setup** We generate 500 samples each for source and target domains using the standard two–moons dataset. The target domain is created by applying a 45° rotation and scaling factor of 1.5 to induce domain shift. All features are standardized prior to adaptation.

**Results** Table 1 reports the classification accuracy on the target domain. Sinkhorn OT improves modestly over the baseline (82.8% → 87.2%), while OT with information loss regularization yields a larger boost to 96.4%. Incorporating cluster-level consistency further enhances the result to 99.8%, nearly eliminating domain mismatch.

Table 1: Classification accuracy (%) on the Two–Moons dataset.

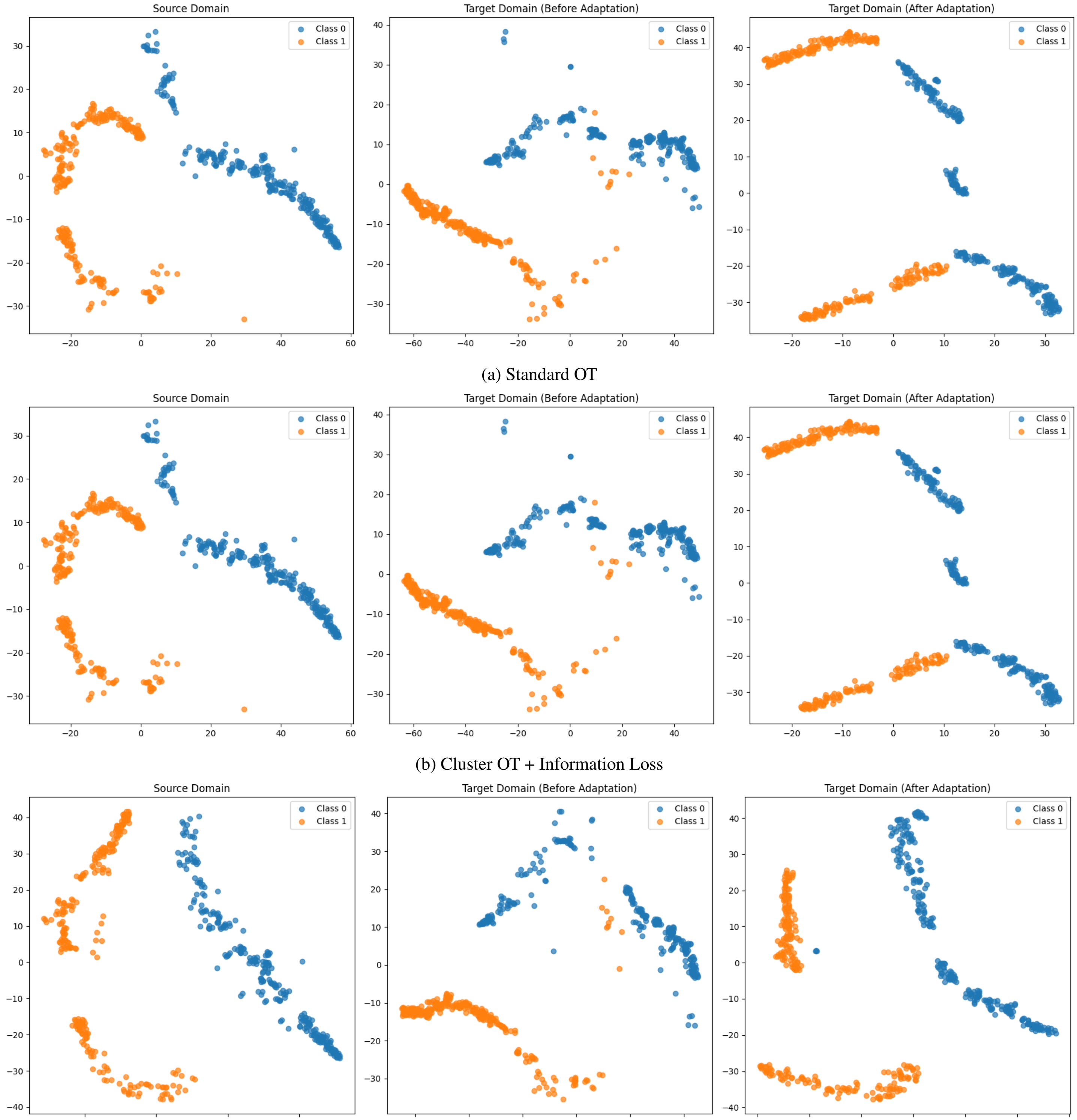
| Method | Target Accuracy |
|---|---|
| No Adaptation | 82.8 |
| Sinkhorn OT | 87.2 |
| Cluster OT + Info–Loss (Ours) | 96.4 |
| OT + Info–Loss (Ours) | **99.8** |

**Visual Analysis** Figure 2 shows t-SNE embeddings of target samples post-adaptation. The inclusion of information loss yields more coherent class clusters. Cluster regularization sharpens separation and aligns semantic structure.

**Semi-Supervised Comparison** Figure 3 compares standard OT and our proposed method (with information loss) under both unsupervised and semi-supervised adaptation settings. Our approach demonstrates consistently better alignment and class separation, even when only 10% of target labels are available.

### 4.2 Real-World: MNIST → USPS

**Setup** We sample 2000 images from MNIST as the source domain. The target domain is generated by transforming a separate set of 2000 MNIST digits with random rotations, scalings, and Gaussian noise to mimic USPS-like characteristics. PCA reduces features to 50 dimensions, followed by standardization.

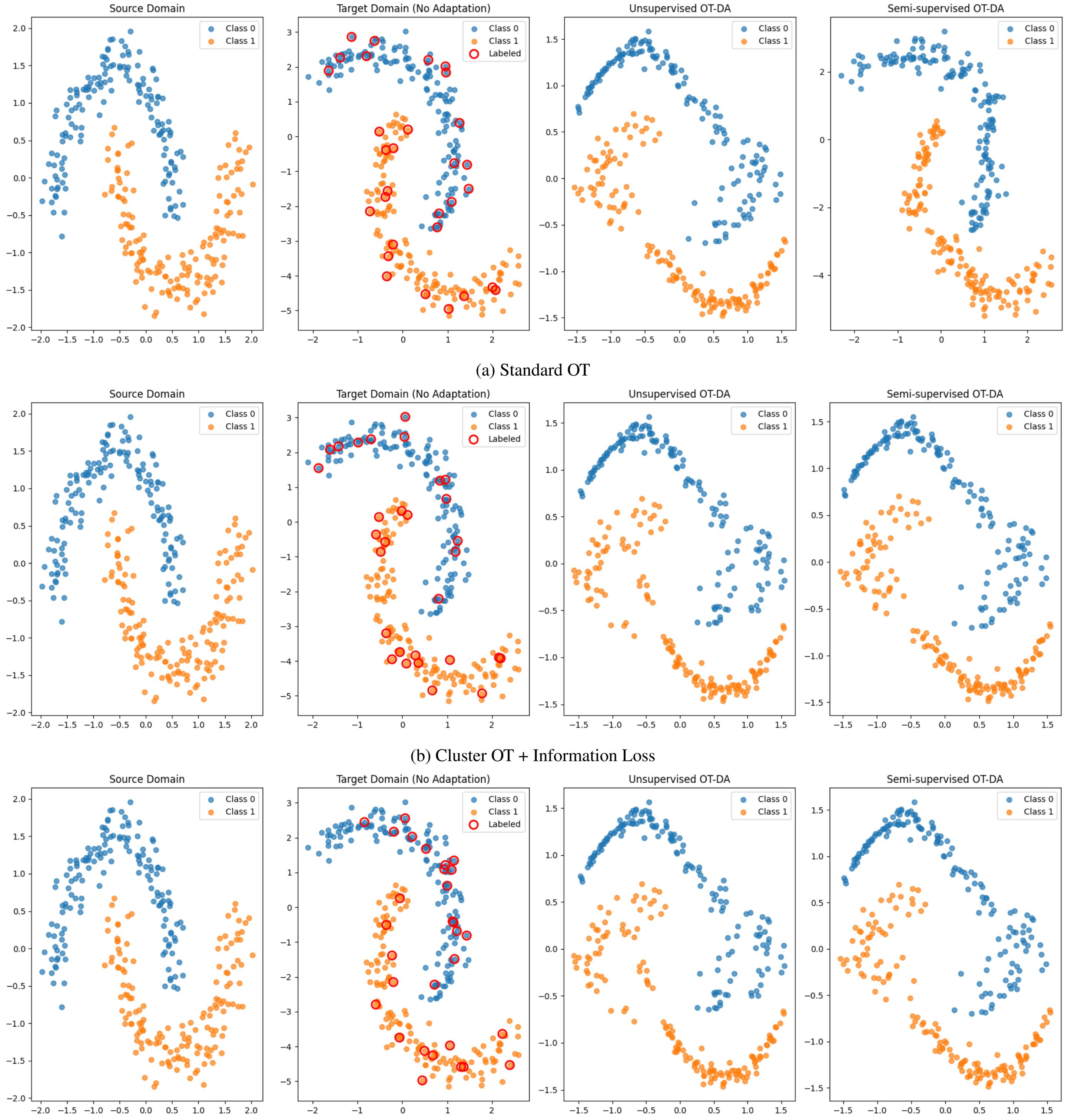


(a) Standard OT

(b) Cluster OT + Information Loss

(c) OT + Information Loss

Figure 2: t-SNE embeddings of Two–Moons target samples after adaptation. Colors indicate true class labels.



(a) Standard OT

(b) Cluster OT + Information Loss

(c) OT + Information Loss

Figure 3: Comparison of standard OT and proposed OT (with information loss) under both unsupervised and semi-supervised domain adaptation settings.

**Results** Table 2 presents accuracy on the target domain. Sinkhorn OT yields a moderate gain over the baseline (71.6% → 77.4%). Our Info–Loss regularized OT improves performance to 80.8%, while adding cluster consistency achieves the best result at 82.7%.

Table 2: Classification accuracy (%) on MNIST → USPS adaptation task.

| Method | Target Accuracy |
|---|---|
| No Adaptation | 71.6 |
| Standard OT | 77.4 |
| Cluster OT + Info–Loss (Ours) | 80.8 |
| OT + Info–Loss (Ours) | **82.7** |

**Visual Analysis** Figure 4 shows t-SNE projections of the adapted embeddings. Our method results in tighter, more discriminative digit clusters in the target space, indicating improved domain alignment.

## 5 Conclusion and Discussion

In this work, we propose a cluster-based transport algorithm that not only first ensures data locality during transports but also shares various merits such as low computational cost and compatibility with any existing optimal transport algorithm. Besides, we propose information loss as a new criterion for selecting the transport. This new loss can only be applied to our cluster-based transport algorithm. Like self-supervised regularization, our information loss term ensures the information preservation assumption holds, but it doesn't need the labels of test data points. We conducted various experiments to show the superiority of our algorithm.

However, there are still some limitations to our work. To be specific, both the method to ensure data locality and the method to ensure information preservation need some improvement. Firstly, although our cluster-based algorithm is simple and effective at ensuring data locality during transport, it is hard to say it is the best way. A better way of ensuring data locality is to learn a neural network that maps any point in the source space to a categorical distribution over target points. In this way, we can ensure data locality with the continuity of our learned deep neural network. Secondly, in our proposed information loss, we measure how differently two clusters transport by measuring the KL divergence between their corresponding categorical distributions over target points. However, the target points are not simply different categories; they have their own geological locations in the target space. We should devise a better way to measure how differently two clusters transport than KL divergence by considering the geological locations of target points.

## 6 Reflection on the progress

Most of the proposed milestones for the project were successfully completed. We derived the theoretical framework and implemented a working prototype of the cluster-based optimal transport algorithm with the proposed information loss term. Initial experiments on synthetic and real-world datasets were conducted, and preliminary results validated our hypothesis about preserving data locality improving domain adaptation.

However, some milestones remain incomplete. In particular, the full suite of experiments across multiple benchmark datasets and the associated statistical analysis could not be fully carried out within the timeline. This was primarily due to the computational cost of running extended experiments and time required for iterative tuning and debugging. Additionally, final refinements to the mathematical model based on empirical feedback are still in progress, contingent on the completion of those experiments.

Despite these pending tasks, the core contributions were implemented and tested, and the project lays a strong foundation for future extensions or more comprehensive evaluation.

## 7 Contribution report

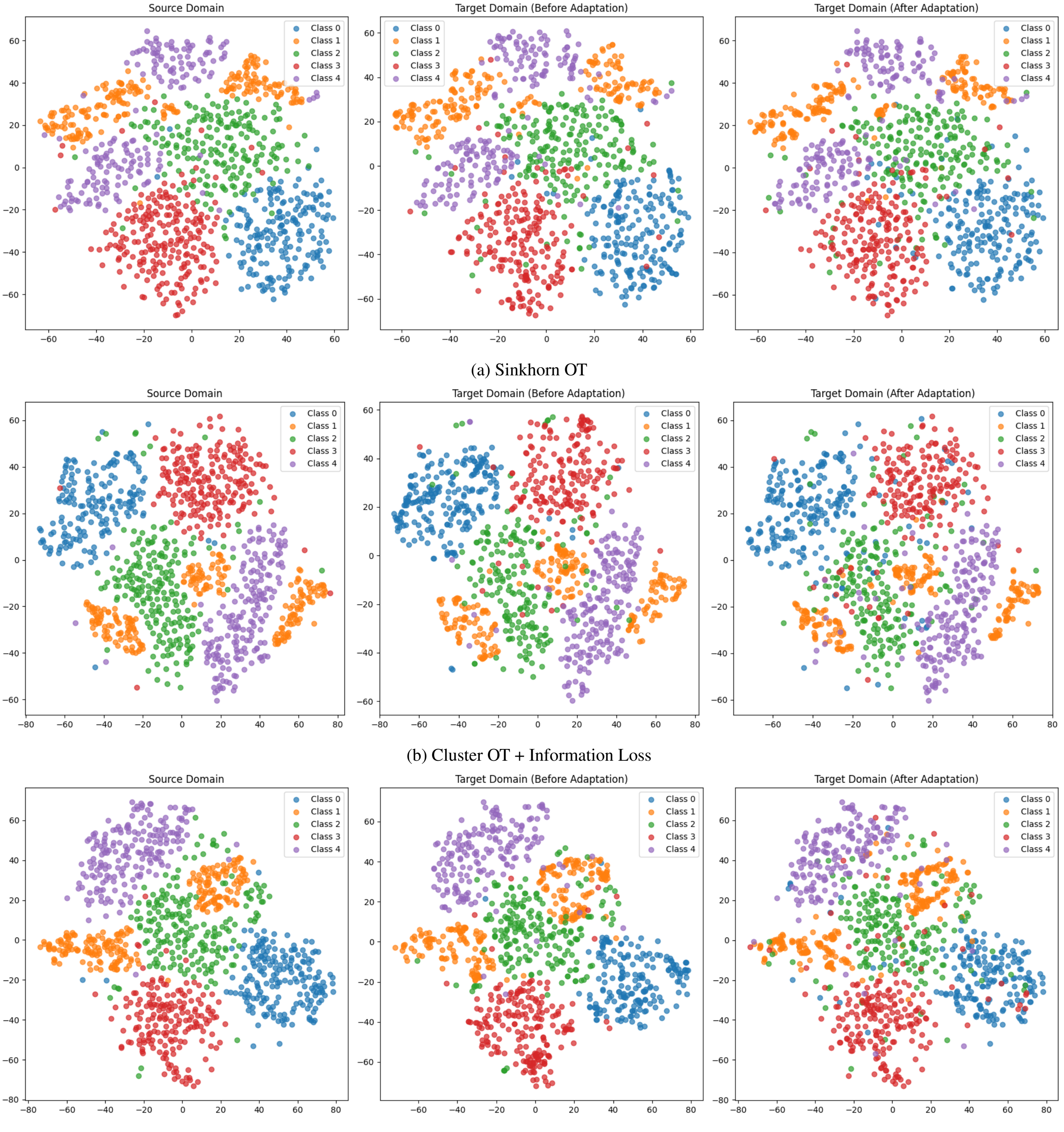
**Rishi's Contributions:** Rishi handled the practical aspects of the project, including data collection, preprocessing, implementation of the proposed method, and execution of experiments. While all major components have been implemented, the comprehensive experimental evaluation and statistical analysis are still in progress due to the computational load and some unexpected implementation hurdles.

**Chuan's Contributions:** Chuan successfully derived the theoretical proofs supporting our core assumption of information preservation, which formed the backbone of our methodology. He also designed the initial algorithm and framework, and took the lead in writing the formal sections of the report. While the refinement of the mathematical model based on empirical data is still ongoing, preliminary updates have been made as results become available.

**Collaborative Milestone:** Weekly meetings were consistently held, serving as a key mechanism for coordination, feedback, and iterative improvement. These sessions helped ensure alignment between the theoretical framework and the empirical implementation.

## 8 Code release

The repository containing the python notebook with the implementation is available at: https://github.com/rishi-more-2003/Improved-Optimal-Transport



(a) Sinkhorn OT

(b) Cluster OT + Information Loss

(c) OT + Information Loss

Figure 4: t-SNE visualizations of transported embeddings for MNIST → USPS. Colors represent digit classes.

## References

[1] C. Villani (2009). *Optimal Transport: Old and New*. Springer-Verlag.

[2] G. Peyré and M. Cuturi (2019). *Computational Optimal Transport*. Foundation and Trends in Machine Learning, 11(5-6):355–607.

[3] Optimal Transport for Domain Adaptation, Nicolas Courty and Rémi Flamary and Devis Tuia and Alain Rakotomamonjy, 2016, https://arxiv.org/abs/1507.00504

[4] Jincheng Yang, Luhao Zhang, Ningyuan Chen, Rui Gao, and Ming Hu. Decision-making with side information: A causal transport robust approach. Optimization Online, 2022.