Appendix

A. Detailed proofs

A.1. Proof of Proposition 1

Proposition 1. *Suppose the latent spaces $\mathcal{Z}_x = \mathcal{Z}_y$ are the same as the original data spaces $\mathcal{X} = \mathcal{Y}$, and the cost matrices are defined by $\mathbf{C}_x[a,b] = \mathbf{C}_z[a,b] = \mathbf{C}_y[a,b] = d(a,b)^p$, where $p \geq 1$ and d is some distance function. If we define the latent Wasserstein discrepancy as $\mathcal{W}_p^L := \left(\mathrm{OT}^{\mathrm{L}}\right)^{1/p}$, then there exist $\kappa > 0$ such that, for any μ, ν and ζ having latent distributions of support sizes up to k, the discrepancy satisfies,*

- $\mathcal{W}_p^L(\mu,\nu) \geq 0$, *(Non-negativity)*
- $\mathcal{W}_p^L(\mu,\nu) = \mathcal{W}_p^L(\nu,\mu)$, *(Symmetry)*
- $\exists \kappa > 0$ *such that* $\mathcal{W}_p^L(\mu,\nu) \leq \kappa \left(\mathcal{W}_p^L(\mu,\zeta) + \mathcal{W}_p^L(\zeta,\nu)\right)$, *(Quasi-Triangle inequality)*

Proof. The first two properties are easy to verify by non-negativity and symmetry of the Wasserstein distance. Hence we will prove the last property. For brevity, we denote $\mathcal{Z} = \mathcal{Z}_x = \mathcal{Z}_y$. Under the assumptions on the cost matrices, we have $\mathrm{OT}_{\mathbf{C}} = \mathcal{W}_p^p$. Hence there exist latent distributions $\mu_z^*, \zeta_z^*, \zeta_z'^*, \nu_z^* \in \Delta_{\mathcal{Z}}^k$ satisfy,

$$(\mathrm{OT}^{\mathrm{L}})(\mu,\zeta) = \mathcal{W}_p^p(\mu,\mu_z^*) + \mathcal{W}_p^p(\mu_z^*,\zeta_z^*) + \mathcal{W}_p^p(\zeta_z^*,\zeta),$$
$$(\mathrm{OT}^{\mathrm{L}})(\zeta,\nu) = \mathcal{W}_p^p(\zeta,\zeta'^*_z) + \mathcal{W}_p^p(\zeta'^*_z,\nu_z^*) + \mathcal{W}_p^p(\nu_z^*,\nu).$$

As d is a distance function, we know that $\mathcal{W}_p$ is a metric (Peyré et al., 2019) that satisfies the triangle inequality:

$$\mathcal{W}_p(\mu,\zeta'^*_z) \leq \mathcal{W}_p(\mu,\mu_z^*) + \mathcal{W}_p(\mu_z^*,\zeta_z^*) + \mathcal{W}_p(\zeta_z^*,\zeta) + \mathcal{W}_p(\zeta,\zeta'^*_z). \tag{13}$$

On the other hand, Jensen's inequality tells us that $(a+b+c+d)^p \leq 4^{p-1}(a^p+b^p+c^p+d^p)$ for any non-negative a,b,c,d. We apply this inequality to (13) and get,

$$\mathcal{W}_p^p(\mu,\zeta'^*_z) \leq 4^{p-1}(\mathcal{W}_p^p(\mu,\mu_z^*) + \mathcal{W}_p^p(\mu_z^*,\zeta_z^*) + \mathcal{W}_p^p(\zeta_z^*,\zeta) + \mathcal{W}_p^p(\zeta,\zeta'^*_z)). \tag{14}$$

Thus,

$$\begin{aligned}
\mathcal{W}_p^L(\mu,\nu) = \left((\mathrm{OT}^{\mathrm{L}})(\mu,\nu)\right)^{\frac{1}{p}} &\leq (\mathcal{W}_p^p(\mu,\zeta'^*_z) + \mathcal{W}_p^p(\zeta'^*_z,\nu_z^*) + \mathcal{W}_p^p(\nu_z^*,\nu))^{\frac{1}{p}} \\
&\overset{(14)}{\leq} 4^{1-\frac{1}{p}}(\mathcal{W}_p^p(\mu,\mu_z^*) + \mathcal{W}_p^p(\mu_z^*,\zeta_z^*) + \mathcal{W}_p^p(\zeta_z^*,\zeta) + \mathcal{W}_p^p(\zeta,\zeta'^*_z) + \mathcal{W}_p^p(\zeta'^*_z,\nu_z^*) + \mathcal{W}_p^p(\nu_z^*,\nu))^{\frac{1}{p}} \\
&\leq 4^{1-\frac{1}{p}}(\mathcal{W}_p^p(\mu,\mu_z^*) + \mathcal{W}_p^p(\mu_z^*,\zeta_z^*) + \mathcal{W}_p^p(\zeta_z^*,\zeta))^{\frac{1}{p}} + (\mathcal{W}_p^p(\zeta,\zeta'^*_z) + \mathcal{W}_p^p(\zeta'^*_z,\nu_z^*) + \mathcal{W}_p^p(\nu_z^*,\nu))^{\frac{1}{p}},
\end{aligned}$$

where the last inequality follows from $(a+b)^{\frac{1}{p}} \leq a^{\frac{1}{p}} + b^{\frac{1}{p}}$ for any nonnegative a,b. Choosing $\kappa = 4^{1-\frac{1}{p}}$ completes the proof. □

A.2. Proof of Corollary 1

Proof. Without ambiguity the optimizations in the followings are over $\Delta_{\mathcal{Z}}^k$. Recall the definition,

$$\tilde{\mathcal{W}}_p^L(\mu,\nu) := \left(\left(\mathcal{W}_p^L(\mu,\nu)\right)^p - \min_{z_k} \mathcal{W}_p^p(\mu,z_k) - \min_{z'_k} \mathcal{W}_p^p(\nu,z'_k)\right)^{1/p} \tag{15}$$

$$= \left(\min_{\mu_z,\nu_z} \left(\mathcal{W}_p^p(\mu,\mu_z) + \mathcal{W}_p^p(\mu_z,\nu_z) + \mathcal{W}_p^p(\nu_z,\nu)\right) - \min_{z_k} \mathcal{W}_p^p(\mu,z_k) - \min_{z'_k} \mathcal{W}_p^p(\nu,z'_k)\right)^{1/p} \tag{16}$$

$$= \left(\left(\mathcal{W}_p^p(\mu,\mu_z^*) - \min_{z_k} \mathcal{W}_p^p(\mu,z_k)\right) + \mathcal{W}_p^p(\mu_z^*,\nu_z^*) + \left(\mathcal{W}_p^p(\nu,\nu_z^*) - \min_{z'_k} \mathcal{W}_p^p(\nu,z'_k)\right)\right)^{1/p} \geq \mathcal{W}_p(\mu_z^*,\nu_z^*) \geq 0, \tag{17}$$

where $(\mu_z^*, \nu_z^*) \in \arg\min_{\mu_z,\nu_z} \big(\mathcal{W}_p^p(\mu,\mu_z) + \mathcal{W}_p^p(\mu_z,\nu_z) + \mathcal{W}_p^p(\nu_z,\nu)\big)$. Observe that the three terms in (17) are non-negative, thus $\tilde{\mathcal{W}}_p^L(\mu,\nu) = 0$ only if the followings are simultaneously satisfied,

$$\mathcal{W}_p^p(\mu,\mu_z^*) = \min_{z_k} \mathcal{W}_p^p(\mu, z_k), \tag{18}$$

$$\mathcal{W}_p^p(\nu,\nu_z^*) = \min_{z_k'} \mathcal{W}_p^p(\nu, z_k'), \tag{19}$$

$$\mathcal{W}_p^p(\mu_z^*,\nu_z^*) = 0. \tag{20}$$

The last condition implies $\mu_z^* = \nu_z^* =: \xi_z$, so the other conditions become $\mathcal{W}_p^p(\mu,\xi_z) = \min_{z_k} \mathcal{W}_p^p(\mu, z_k)$ and $\mathcal{W}_p^p(\mu,\xi_z) = \min_{z_k'} \mathcal{W}_p^p(\nu, z_k')$. We recognize that $\min_{z_k} \mathcal{W}_p^p(\mu, z_k)$ is a Wasserstein barycenter problem such that the barycenter supports on up to k locations. So the conditions imply that the barycenters of μ and ν must coincide. On the other hand, when $p = 2$, (Peyré et al., 2019; Ho et al., 2017) show that the solution z_k to the barycenter problem is the distribution of k-means centroids of μ with a distribution proportional to the mass of clusters. The same conclusion also applies to ν. Hence, the condition (20) shows that not only the centroids of k-means to μ and ν must be the same, but also the proportions of their corresponding clusters must be equal, which completes the proof of the corollary. □

A.3. Proof of Proposition 2

The proof of Proposition 2 will rely on the following lemma.

Lemma 1. *(Log-sum inequality) : Let x_m and y_m, $m = 1,\dots,n$, be nonnegative sequences, then*

$$\left(\sum_{m=1}^n x_m\right) \log\left(\frac{\sum_{m=1}^n x_m}{\sum_{m=1}^n y_m}\right) \le \sum_{m=1}^n x_m \log\left(\frac{x_m}{y_m}\right). \tag{21}$$

Proof. Denote $x = \sum_m x_m$ and $y = \sum_m y_m$. By concavity of the $\log$ function and Jensen's inequality, we have,

$$\sum_m \frac{x_m}{x} \log\left(\frac{y_m}{x_m}\right) \le \log\left(\sum_m \frac{x_m}{x}\frac{y_m}{x_m}\right) = \log\left(\frac{y}{x}\right). \tag{22}$$

Multiplying both sides of the above inequality with $-x$ yields the lemma. □

Proposition 2. *Let $\mathbf{P}$ be transport plan of the form in (5). Assume $\mathbf{K}$ is some Gibbs kernel satisfying $\mathbf{K}_x\mathbf{K}_z\mathbf{K}_y \le \mathbf{K}$, where the inequality is over each entry. The following inequality holds,*

$$\varepsilon KL(\mathbf{P}\|\mathbf{K}) \le \varepsilon(KL(\mathbf{P}_x\|\mathbf{K}_x) + KL(\mathbf{P}_z\|\mathbf{K}_z) + KL(\mathbf{P}_y\|\mathbf{K}_y) + \mathbf{H}(\mathbf{u}_z) + \mathbf{H}(\mathbf{v}_z)), \tag{23}$$

where $\mathbf{H}(\mathbf{a}) := -\sum_i \mathbf{a}_i \log \mathbf{a}_i$ denotes the Shannon entropy.

Proof. As the transport cost is monotonically decreasing in the entries of $\mathbf{K}$, we only need to prove the case when $\mathbf{K}_x\mathbf{K}_z\mathbf{K}_y = \mathbf{K}$. To simplify notations, we define $\tilde{\mathbf{P}}_z := \mathrm{diag}(\mathbf{u}_z^{-1})\mathbf{P}_z\mathrm{diag}(\mathbf{v}_z^{-1})$, $\tilde{\mathbf{P}}_y := \mathrm{diag}(\mathbf{v}_z^{-1})\mathbf{P}_y$. Now by definition of $\mathbf{P}$ with the form (5) we have,

$$\mathrm{KL}(\mathbf{P}\|\mathbf{K}) = \sum_{i,j}\left(\sum_m \left((\mathbf{P}_x)_{i,m}(\tilde{\mathbf{P}}_z\mathbf{P}_y)_{m,j}\right) \log \frac{\sum_m (\mathbf{P}_x)_{i,m}(\tilde{\mathbf{P}}_z\mathbf{P}_y)_{m,j}}{\sum_m (\mathbf{K}_x)_{i,m}(\mathbf{K}_z\mathbf{K}_y)_{m,j}}\right) =: \sum_{i,j} a_{i,j}. \tag{24}$$

Apply Lemma 1 to $a_{i,j}$ with $x_m = (\mathbf{P}_x)_{i,m}(\tilde{\mathbf{P}}_z\mathbf{P}_y)_{m,j}$ and $y_m = \sum_m (\mathbf{K}_x)_{i,m}(\mathbf{K}_z\mathbf{K}_y)_{m,j}$ we have,

$$a_{i,j} \le \sum_m (\mathbf{P}_x)_{i,m}(\tilde{\mathbf{P}}_z\mathbf{P}_y)_{m,j} \log\frac{(\mathbf{P}_x)_{i,m}}{(\mathbf{K}_x)_{i,m}} + \sum_m (\mathbf{P}_x)_{i,m}(\tilde{\mathbf{P}}_z\mathbf{P}_y)_{m,j} \log\frac{(\tilde{\mathbf{P}}_z\mathbf{P}_y)_{m,j}}{(\mathbf{K}_z\mathbf{K}_y)_{m,j}} =: b_{i,j} + c_{i,j}. \tag{25}$$

Since $\tilde{\mathbf{P}}_z\mathbf{P}_y\mathbf{1} = \mathrm{diag}(\mathbf{u}_z^{-1})\mathbf{P}_z\mathrm{diag}(\mathbf{v}_z^{-1})\mathbf{v}_z = \mathrm{diag}(\mathbf{u}_z^{-1})\mathbf{u}_z = \mathbf{1}$, we have $\sum_j (\tilde{\mathbf{P}}_z\mathbf{P}_y)_{m,j} = 1$, $\forall m$, and

$$\sum_{i,j} b_{i,j} = \sum_i \sum_m (\mathbf{P}_x)_{i,m}\left(\sum_j (\tilde{\mathbf{P}}_z\mathbf{P}_y)_{m,j}\right) \log\frac{(\mathbf{P}_x)_{i,m}}{(\mathbf{K}_x)_{i,m}} = \sum_{i,m} (\mathbf{P}_x)_{i,m} \log\frac{(\mathbf{P}_x)_{i,m}}{(\mathbf{K}_x)_{i,m}} = \mathrm{KL}(\mathbf{P}_x\|\mathbf{K}_x). \tag{26}$$

On the other hand, $\mathbf{P}_x^T\mathbf{1} = \mathbf{u}_z$ implies $\sum_i(\mathbf{P}_x)_{i,m} = (\mathbf{u}_z)_m$, $\forall m$, and

$$\begin{aligned}\sum_{i,j} c_{i,j} &= \sum_{m,j}(\mathbf{u}_z)_m(\tilde{\mathbf{P}}_z\mathbf{P}_y)_{m,j}\log\frac{(\tilde{\mathbf{P}}_z\mathbf{P}_y)_{m,j}}{(\mathbf{K}_z\mathbf{K}_y)_{m,j}} = \sum_{m,j}(\mathbf{P}_z\mathrm{diag}(\mathbf{v}_z^{-1})\mathbf{P}_y)_{m,j}\log\frac{(\mathbf{P}_z\mathrm{diag}(\mathbf{v}_z^{-1})\mathbf{P}_y)_{m,j}}{(\mathbf{u}_z)_m(\mathbf{K}_z\mathbf{K}_y)_{m,j}}\\ &\overset{(i)}{=}\mathbf{H}(\mathbf{u}_z) + \sum_{m,j}(\mathbf{P}_z\mathrm{diag}(\mathbf{v}_z^{-1})\mathbf{P}_y)_{m,j}\log\frac{(\mathbf{P}_z\mathrm{diag}(\mathbf{v}_z^{-1})\mathbf{P}_y)_{m,j}}{(\mathbf{K}_z\mathbf{K}_y)_{m,j}}\\ &\overset{(ii)}{=}\mathbf{H}(\mathbf{u}_z) + \sum_{m,j}(\mathbf{P}_z\tilde{\mathbf{P}}_y)_{m,j}\log\frac{(\mathbf{P}_z\tilde{\mathbf{P}}_y)_{m,j}}{(\mathbf{K}_z\mathbf{K}_y)_{m,j}} =: \mathbf{H}(\mathbf{u}_z) + \sum_{m,j} d_{m,j},\end{aligned} \tag{27}$$

where (i) follows from $\mathbf{P}_z\mathrm{diag}(\mathbf{v}_z^{-1})\mathbf{P}_y\mathbf{1} = \mathbf{u}_z$ and (ii) from the definition of $\tilde{\mathbf{P}}_y$. Now applying Lemma 1 again to $d_{m,j}$ with $x_l = (\mathbf{P}_z)_{m,l}(\tilde{\mathbf{P}}_y)_{l,j},\ y_l = (\mathbf{K}_z)_{m,l}(\mathbf{K}_y)_{l,j}$ leads to

$$\begin{aligned}d_{m,j} &\le \sum_l\Big((\mathbf{P}_z)_{m,l}(\tilde{\mathbf{P}}_y)_{l,j}\Big)\log\frac{(\mathbf{P}_z)_{m,l}(\tilde{\mathbf{P}}_y)_{l,j}}{(\mathbf{K}_z)_{m,l}(\mathbf{K}_y)_{l,j}}\\ &= \sum_l\Big((\mathbf{P}_z)_{m,l}(\tilde{\mathbf{P}}_y)_{l,j}\Big)\log\frac{(\mathbf{P}_z)_{m,l}}{(\mathbf{K}_z)_{m,l}} + \sum_l\Big((\mathbf{P}_z)_{m,l}(\tilde{\mathbf{P}}_y)_{l,j}\Big)\log\frac{(\tilde{\mathbf{P}}_y)_{l,j}}{(\mathbf{K}_y)_{l,j}} =: e_{m,j} + f_{m,j}.\end{aligned} \tag{28}$$

From $\tilde{\mathbf{P}}_y\mathbf{1} = \mathrm{diag}(\mathbf{v}_z^{-1})\mathbf{P}_y\mathbf{1} = (\mathbf{v}_z^{-1})\mathbf{v}_z = \mathbf{1}$, we get $\sum_j(\tilde{\mathbf{P}}_y)_{l,j} = 1$ and

$$\sum_{m,j} e_{m,j} = \sum_m(\mathbf{P}_z)_{m,l}\sum_j\Big((\tilde{\mathbf{P}}_y)_{l,j}\Big)\log\frac{(\mathbf{P}_z)_{m,l}}{(\mathbf{K}_z)_{m,l}} = \mathrm{KL}(\mathbf{P}_z\|\mathbf{K}_z). \tag{29}$$

Similarly, using $\mathbf{P}_z^T\mathbf{1} = \mathbf{v_z}$, we also get $\sum_m(\mathbf{P}_z)_{m,l} = (\mathbf{v}_z)_l$, $\forall l$, and

$$\begin{aligned}\sum_{m,j} f_{mj} &= -\sum_{m,j,l}(\mathbf{P}_z)_{m,l}(\tilde{\mathbf{P}}_y)_{l,j}\log(\mathbf{v}_z)_l + \sum_{m,j,l}(\mathbf{P}_z)_{m,l}(\tilde{\mathbf{P}}_y)_{l,j}\log\frac{(\mathbf{P}_y)_{l,j}}{(\mathbf{K}_y)_{l,j}}\\ &= \mathbf{H}(\mathbf{v}_z) + \mathrm{KL}(\mathbf{P}_y\|\mathbf{K}_y),\end{aligned} \tag{30}$$

where the last equality follows from $\sum_j(\tilde{\mathbf{P}}_y)_{l,j} = 1$ and $\sum_m(\mathbf{P}_z)_{m,l}(\tilde{\mathbf{P}}_y)_{l,j} = (\mathbf{v}_z)_l(\mathbf{v}_z)_l^{-1}(\mathbf{P}_y)_{l,j}$. Finally, combining equations from (24-30) completes the proof. □

A.4. Proof of Corollary 2

Proof. By the assumptions on the cost matrices, the condition $\mathbf{K}_x\mathbf{K}_z\mathbf{K}_y \le \mathbf{K}$ is equivalent to

$$\|\mathbf{x}_i - \mathbf{y}_j\|_p^p \le -\varepsilon\log\left(\sum_{l,m}\exp\left(\frac{-1}{\varepsilon}\left(3^{p-1}\|\mathbf{x}_i - \mathbf{z}_l^x\|_p^p + 3^{p-1}\|\mathbf{z}_l^x - \mathbf{z}_m^y\|_p^p + 3^{p-1}\|y_j - \mathbf{z}_m^y\|_p^p\right)\right)\right). \tag{31}$$

Observe on the right-hand side is nothing more than the soft-min of the set

$$\mathcal{A} := \{3^{p-1}\|\mathbf{x}_i - \mathbf{z}_l^x\|_p^p + 3^{p-1}\|\mathbf{z}_l^x - \mathbf{z}_m^y\|_p^p + 3^{p-1}\|y_j - \mathbf{z}_m^y\|_p^p : \forall l, m\}, \tag{32}$$

where $\text{soft-min}_\varepsilon(\mathcal{A}) := -\varepsilon\log(\sum_{\alpha\in\mathcal{A}}\exp(-\alpha/\varepsilon))$, which approaches the minimum of the set $\min\mathcal{A}$ as $\varepsilon\to 0$. On the other hand, any element in the set is at least $\|\mathbf{x}_i - \mathbf{y}_j\|_p^p$ because

$$\begin{aligned}\|\mathbf{x}_i - \mathbf{y}_j\|_p^p &\overset{(i)}{\le} (\|\mathbf{x}_i - \mathbf{z}_l^x\|_p + \|\mathbf{z}_l^x - \mathbf{z}_m^y\|_p + \|\mathbf{y}_j - \mathbf{z}_m^y\|_p)^p\\ &\overset{(ii)}{\le} 3^{p-1}(\|\mathbf{x}_i - \mathbf{z}_l^x\|_p + \|\mathbf{z}_l^x - \mathbf{z}_m^y\|_p + \|\mathbf{y}_j - \mathbf{z}_m^y\|_p),\end{aligned} \tag{33}$$

where (i) follows from Minkowski inequality and (ii) from applying Jensen's inequality to convex function x^p. Hence, (31) holds for sufficient small ε, and thus `LOT` with these cost matrices gives a transport plan according to the upper bound in proposition 2. □

A.5. Proof of Proposition 3

Proposition 3. *Suppose X and Y have distributions μ and ν supported on a compact region Ω in $\mathbb{R}^d$, the cost functions $c_x(\cdot,\cdot)$ and $c_y(\cdot,\cdot)$ are defined as the squared Euclidean distance, and $\hat{\mu}$, $\hat{\nu}$ are empirical distributions of n and m i.i.d. samples from μ and ν, respectively. If the spaces for latent distributions are equal to $\mathcal{Z}_x = \mathcal{Z}_y = \mathbb{R}^d$, and there are k_x and k_y anchors in the source and target, respectively, then with probability at least $1-\delta$,*

$$\text{Err} \ \leq C\sqrt{\frac{k_{max}^3 d \log k_{max} + \log(2/\delta)}{N}}, \tag{34}$$

where $\text{Err} = |\text{OT}^L(\mu,\nu) - \text{OT}^L(\hat{\mu},\hat{\nu})|$, $k_{max} = \max\{k_x, k_y\}$ *and* $N = \min\{n, m\}$ *and* $C \geq 0$ *is some constant not depending on* N.

The proof is an application of the following Lemma.

Lemma 2. *(Theorem 4 in (Forrow et al., 2019)). Suppose μ is a distribution supported in a compact region $K \subseteq \mathbb{R}^d$. Let $\hat{\mu}$ be an empirical distribution of n i.i.d. samples from μ, then there exists a constant $C > 0$ such that for any distribution μ_z supported on up to k points, with probability $1-\delta$, we have,*

$$|\mathcal{W}_2^2(\mu,\mu_z) - \mathcal{W}_2^2(\hat{\mu},\mu_z)| \leq C\sqrt{\frac{k^3 d \log k + \log(2/\delta)}{n}}. \tag{35}$$

Proof. (of Proposition 3). Following the assumptions of the proposition, $OT^L(\mu,\nu)$, $OT^L(\hat{\mu},\hat{\nu})$ can be rewritten into optimizations involving the 2-Wasserstein distance,

$$\text{OT}^L(\mu,\nu) = \min_{\mu_z \in \Delta_{\mathbb{R}^d}^{k_x}, \nu_z \in \Delta_{\mathbb{R}^d}^{k_y}} \mathcal{W}_2^2(\mu,\mu_z) + \mathcal{W}_2^2(\mu_z,\nu_z) + \mathcal{W}_2^2(\nu_z,\nu), \tag{36}$$

$$\text{OT}^L(\hat{\mu},\hat{\nu}) = \min_{\mu_z \in \Delta_{\mathbb{R}^d}^{k_x}, \nu_z \in \Delta_{\mathbb{R}^d}^{k_y}} \mathcal{W}_2^2(\hat{\mu},\mu_z) + \mathcal{W}_2^2(\mu_z,\nu_z) + \mathcal{W}_2^2(\nu_z,\hat{\nu}). \tag{37}$$

Without lose of generality, we assume that $\text{OT}^L(\mu,\nu) \geq \text{OT}^L(\hat{\mu},\hat{\nu})$. Let (μ_z^*, ν_z^*) and $(\hat{\mu}_z^*, \hat{\nu}_z^*)$ denote the optimal solutions to the two optimization problems. Then,

$$\begin{aligned}
&\text{OT}^L(\mu,\nu) - \text{OT}^L(\hat{\mu},\hat{\nu}) \\
&= \mathcal{W}_2^2(\mu,\mu_z^*) + \mathcal{W}_2^2(\mu_z^*,\nu_z^*) + \mathcal{W}_2^2(\nu_z^*,\nu) - \big(\mathcal{W}_2^2(\hat{\mu},\hat{\mu}_z^*) + \mathcal{W}_2^2(\hat{\mu}_z^*,\hat{\nu}_z^*) + \mathcal{W}_2^2(\hat{\nu}_z^*,\hat{\nu})\big) \\
&\overset{(a)}{\leq} \mathcal{W}_2^2(\mu,\hat{\mu}_z^*) + \mathcal{W}_2^2(\hat{\mu}_z^*,\hat{\nu}_z^*) + \mathcal{W}_2^2(\hat{\nu}_z^*,\nu) - \big(\mathcal{W}_2^2(\hat{\mu},\hat{\mu}_z^*) + \mathcal{W}_2^2(\hat{\mu}_z^*,\hat{\nu}_z^*) + \mathcal{W}_2^2(\hat{\nu}_z^*,\hat{\nu})\big) \\
&\leq |\mathcal{W}_2^2(\mu,\hat{\mu}_z^*) - \mathcal{W}_2^2(\hat{\mu},\hat{\mu}_z^*)| + |\mathcal{W}_2^2(\nu,\hat{\nu}_z^*) - \mathcal{W}_2^2(\hat{\nu},\hat{\nu}_z^*)| \\
&\overset{(b)}{\leq} C_1\sqrt{\frac{k_x^3 d \log k_x + \log(2/\delta)}{n}} + C_2\sqrt{\frac{k_y^3 d \log k_y + \log(2/\delta)}{m}} \\
&\leq (C_1 + C_2)\sqrt{\frac{k_{\max}^3 d \log k_{\max} + \log(2/\delta)}{N}},
\end{aligned}$$

where (a) follows from the optimality of (μ_z^*, ν_z^*) to the objective (36), (b) by applying Lemma 1 twice. By symmetry, we get a similar result for the case where $\text{OT}^L(\mu,\nu) \leq \text{OT}^L(\hat{\mu},\hat{\nu})$ which completes the proof. □

B. Additional algorithms and derivations

B.1. Derivation of Algorithm 1

In this section, we provide the detailed derivation of the algorithm 1. It is based on the Dykstra's algorithm (Dykstra, 1983) which considers problems of the form:

$$\begin{aligned}
&\min_{\mathbf{P} \in \mathbf{R}^{n \times m}} \text{KL}(\mathbf{P}||\mathbf{K}) \\
&\text{subject to } \mathbf{P} \in \bigcap_{i=1}^{k} \mathcal{C}_i,
\end{aligned} \tag{38}$$

Algorithm 2 Iterative Bregman Projection

Input: $\mathbf{K}$
Output: $\mathbf{P}$
Initialize: $\mathbf{P} \leftarrow \mathbf{K}$
while not converging **do**
  for $i = 1, \dots, k$ **do**
    $\mathbf{P} \leftarrow \Pi_{\mathcal{C}_i}^{\mathrm{KL}}(\mathbf{P})$
  end for
end while

where $\mathbf{K} \in \mathbb{R}_+^{n \times m}$ is some non-negative fixed matrix and $\mathcal{C}_i, \forall i$ are closed convex sets. When in addition $\mathcal{C}_i$ are affine, an iterative Bregman projection (Benamou et al., 2015) solves the problem and has the form of algorithm 2. Here $\Pi_{\mathcal{C}}^{\mathrm{KL}}(\mathbf{P})$ denotes the Bregman projection of $\mathbf{P}$ onto $\mathcal{C}$ defined as $\Pi_{\mathcal{C}}^{\mathrm{KL}}(\mathbf{P}) := \mathrm{argmin}_{\gamma \in \mathcal{C}} \mathrm{KL}(\gamma || \mathbf{P})$.

Recall that the objective of algorithm 1 is

$$\begin{aligned} &\min_{\mathbf{P}_x, \mathbf{P}_z, \mathbf{P}_y} \varepsilon(\mathrm{KL}(\mathbf{P}_x \| \mathbf{K}_x) + \mathrm{KL}(\mathbf{P}_z \| \mathbf{K}_z) + \mathrm{KL}(\mathbf{P}_y \| \mathbf{K}_y)), \\ &\text{subject to: } \exists \mathbf{u}_z, \mathbf{v}_z : \mathbf{P}_x \mathbf{1} = \mu, \mathbf{P}_x^T \mathbf{1} = \mathbf{u}_z, \mathbf{P}_z \mathbf{1} = \mathbf{u}_z, \mathbf{P}_z^T \mathbf{1} = \mathbf{v}_z, \mathbf{P}_y \mathbf{1} = \mathbf{v}_z, \mathbf{P}_y^T \mathbf{1} = \nu. \end{aligned} \tag{39}$$

Hence, we can write (39) into the form of (38) by defining

$$\begin{aligned} &\mathcal{C}_1 = \{\mathbf{P}_x : \mathbf{P}_x \mathbf{1} = \mu\}, \mathcal{C}_2 = \{\mathbf{P}_y : \mathbf{P}_y^T \mathbf{1} = \nu\}, \\ &\mathcal{C}_3 = \{(\mathbf{P}_x, \mathbf{P}_z) : \exists \mathbf{v}_z, \mathbf{P}_x^T \mathbf{1} = \mathbf{P}_z \mathbf{1} = \mathbf{u}_z\}, \mathcal{C}_4 = \{(\mathbf{P}_z, \mathbf{P}_y) : \exists \mathbf{u}_z, \mathbf{P}_z^T \mathbf{1} = \mathbf{P}_y \mathbf{1} = \mathbf{v}_z\}. \end{aligned} \tag{40}$$

For further simplifications, observe that each transport matrix is a Bregman projection per iteration, so it can be easily verified that the transport plan must be of the form $\mathbf{P}_i = \mathrm{diag}(\alpha_i) \mathbf{K}_i \mathrm{diag}(\beta_i)$, for some $\alpha_i \in \mathbb{R}_+^n, \beta_i \in \mathbb{R}_+^m$, $i \in \{x, y, z\}$. Now, using a standard technique of Lagrange multipliers, the projections to each of these sets (40) can be written as

$$\begin{aligned} &\Pi_{\mathcal{C}_1}^{\mathrm{KL}}(\mathbf{P}_x) = \mathrm{diag}(\mu \oslash \mathbf{K}_x \beta) \mathbf{K}_x \mathrm{diag}(\beta_x), \\ &\Pi_{\mathcal{C}_2}^{\mathrm{KL}}(\mathbf{P}_y) = \mathrm{diag}(\alpha_y) \mathbf{K}_y \mathrm{diag}(\nu \oslash \mathbf{K}_y^T \alpha_y), \\ &\Pi_{\mathcal{C}_3}^{\mathrm{KL}}(\mathbf{P}_x) = \mathrm{diag}(\alpha_x) \mathbf{K}_x \mathrm{diag}(\mathbf{u}_z \oslash \mathbf{K}_x^T \alpha_x),\ \Pi_{\mathcal{C}_3}^{\mathrm{KL}}(\mathbf{P}_z) = \mathrm{diag}(\mathbf{u}_z \oslash \mathbf{K}_z \beta_z) \mathbf{K}_x \mathrm{diag}(\beta_z), \\ &\text{where } \mathbf{u}_z = ((\alpha_z \odot \mathbf{K}_z \beta_z) \odot (\beta_x \odot \mathbf{K}_x^T \alpha_x))^{\frac{1}{2}} \\ &\Pi_{\mathcal{C}_4}^{\mathrm{KL}}(\mathbf{P}_z) = \mathrm{diag}(\alpha_z) \mathbf{K}_z \mathrm{diag}(\mathbf{v}_z \oslash \mathbf{K}_z^T \alpha_z),\ \Pi_{\mathcal{C}_4}^{\mathrm{KL}}(\mathbf{P}_y) = \mathrm{diag}(\mathbf{v}_z \oslash \mathbf{K}_y \beta_y) \mathbf{K}_y \mathrm{diag}(\beta_y), \\ &\text{where } \mathbf{v}_z = ((\alpha_y \odot \mathbf{K}_y \beta_y) \odot (\beta_z \odot \mathbf{K}_z^T \alpha_z))^{\frac{1}{2}}. \end{aligned}$$

Finally, by keeping track with α_i, β_i, we arrive at the algorithm 1.

B.2. Rule of anchor update

When $\mathbf{C}_x = [d_{\mathbf{M}_x}(\mathbf{x}_i, \mathbf{z}_m^x)]_{i,m}, \mathbf{C}_z := [d_{\mathbf{M}_z}(\mathbf{z}_m^x, \mathbf{z}_n^y)]_{m,n}, \mathbf{C}_y = [d_{\mathbf{M}_y}(\mathbf{x}_j, \mathbf{z}_n^y)]_{n,j}$, we can rewrite the objective of `LOT` explicitly as a function of $\mathbf{Z}_x$ and $\mathbf{Z}_y$:

$$\begin{aligned} \mathrm{OT}^{\mathrm{L}} =& \mathrm{tr}\left(\mathbf{Z}_x^T (\mathbf{M}_x + \mathbf{M}_y) \mathbf{Z}_x \mathrm{diag}(\mathbf{u}_z)\right) + \mathrm{tr}\left(\mathbf{Z}_y^T (\mathbf{M}_y + \mathbf{M}_z) \mathbf{Z}_y \mathrm{diag}(\mathbf{v}_z)\right) \\ &- 2\mathrm{tr}\left(\mathbf{P}_x^T \mathbf{X}^T \mathbf{M}_x \mathbf{Z}_x\right) - 2\mathrm{tr}\left(\mathbf{P}_z^T \mathbf{Z}_x^T \mathbf{M}_z \mathbf{Z}_y\right) - 2\mathrm{tr}(\mathbf{P}_y^T \mathbf{Z}_y^T \mathbf{M}_y \mathbf{Y}). \end{aligned} \tag{41}$$

We get the first-order stationary point by setting the gradient of OT^{L} with respect to $\mathbf{Z}_x$ and $\mathbf{Z}_y$ to zero, which results in the closed-form formula,

$$\begin{bmatrix} \mathrm{vec}(\mathbf{Z}_x^*) \\ \mathrm{vec}(\mathbf{Z}_y^*) \end{bmatrix} = \begin{bmatrix} \mathrm{diag}(\mathbf{u}_z) \otimes (\mathbf{M}_x + \mathbf{M}_z) & \mathbf{P}_z \otimes \mathbf{M}_z \\ -\mathbf{P}_z^T \otimes \mathbf{M}_z & \mathrm{diag}(\mathbf{v}_z) \otimes (\mathbf{M}_y + \mathbf{M}_z) \end{bmatrix}^{-1} \begin{bmatrix} (\mathbf{P}_x^T \otimes \mathbf{M}_x) \mathrm{vec}(\mathbf{X}) \\ (\mathbf{P}_y \otimes \mathbf{M}_y) \mathrm{vec}(\mathbf{Y}) \end{bmatrix}. \tag{42}$$

Based on the formula, we present a Floyd-type algorithm in Algorithm 1 by alternatively the optimization of transport plans $(\mathbf{P}_x, \mathbf{P}_y, \mathbf{P}_z)$ and the optimization of anchors $(\mathbf{Z}_x, \mathbf{Z}_y)$.

Algorithm 3 Latent Optimal Transport - Wasserstein Ground Metric (LOT-WA)

Input: $\mathbf{P}_x, \mathbf{P}_y, \mathbf{P}_z, \mathbf{P}_{in}, \mathbf{K}_x, \mathbf{K}_y, \theta$.
Output: $\mathbf{P}_x, \mathbf{P}_y, \mathbf{P}_z, \mathbf{P}_{in}, \mathbf{Z}_x, \mathbf{Z}_y$

while not converging **do**
  for $m = 1, \ldots, k_x$ **do**
    for $n = 1, \ldots, k_y$ **do**
      if $\mathbf{P}_z(\mathbf{z}_m^x, \mathbf{z}_n^y) > \theta \max_n \mathbf{P}_z(\mathbf{z}_m^x, \mathbf{z}_n^y)$ **then**
        $\mathbf{C}_z[m,n], \mathbf{P}_{in}[m,n] =$ SOLVEOT $(\mathbf{P}_x(\cdot|\mathbf{z}_m^x), \mathbf{P}_y(\cdot|\mathbf{z}_n^y))$ {//calculate $\mathcal{W}_2^2(\mathbf{P}_x(\cdot|\mathbf{z}_m^x), \mathbf{P}_y(\cdot|\mathbf{z}_n^y))$}
      else
        $\mathbf{C}_z[m,n] = \infty$
      end if
    end for
  end for
  $\mathbf{P}_x, \mathbf{P}_y, \mathbf{P}_z \leftarrow$ UPDATEPLAN$(\mathbf{K}_x, \mathbf{K}_y, \exp(-\mathbf{C}_z/\varepsilon))$
end while

B.3. Wasserstein distance as ground metric and estimation of transported points

This section we consider the inner cost matrix to be defined by the Wasserstein distances of the clustered distributions as $\mathbf{C}_z := [\mathcal{W}_2^2(\mathbf{P}_x(\cdot|\mathbf{z}_m^x), \mathbf{P}_y(\cdot|\mathbf{z}_n^y))]_{m,n}$. This form arises when we represent the centroids by measures of points as $\tilde{\mathbf{z}}^x = \sum_{i=1}^N \mathbf{P}_x(\mathbf{x}_i|\mathbf{z}^x)\delta_{\mathbf{x}_i}$, $\tilde{\mathbf{z}}^y = \sum_{j=1}^M \mathbf{P}_y(\mathbf{y}_j|\mathbf{z}^y)\delta_{\mathbf{x}_j}$. Intuitively, $\tilde{\mathbf{z}}^x$ represents the distribution of the source points associated with the anchor $\mathbf{z}^x$ while $\tilde{\mathbf{z}}^y$ represents that of the target points associated with the anchor $\mathbf{z}^y$. We thus measure the distance between the anchors using the minimal transportation cost between the two distributions, which is then equivalent to the Wasserstein distance between the two measures. The additional challenges in optimizing LOT here is that the inner cost matrix depends on the transport plans of outer OTs. We propose to use an alternating optimization scheme that optimize transports plans $(\mathbf{P}_x, \mathbf{P}_y, \mathbf{P}_z)$ while keeping fixed $\mathbf{C}_z$ and then optimize $\mathbf{C}_z$ while keeping $(\mathbf{P}_x, \mathbf{P}_y, \mathbf{P}_z)$ fixed. However, the computation complexity of $\mathbf{C}_z$ could be high as it requires solving $k_x k_y$ OTs per iteration, where k_x, k_y are the numbers of anchors in the source and target, respectively. We can reduce the computation by only solving a few OTs corresponding to anchor pairs $(m,n) \in [k_x] \times [k_y]$ that have high transportation probabilities $\mathbf{P}_z$. Specifically, for each $(m,n) \in [k_x] \times [k_y]$, we adopt the criteria of only calculating $\mathcal{W}_2^2(\mathbf{P}_x(\cdot|\mathbf{z}_m^x), \mathbf{P}_y(\cdot|\mathbf{z}_n^y))$ when $\mathbf{P}_z(\mathbf{z}_m^x, \mathbf{z}_n^y) > \theta \max_n \mathbf{P}_z(\mathbf{z}_m^x, \mathbf{z}_n^y)$, where θ is some predefined threshold (we use $\theta = 0.5$ in the experiment). For the uncalculated Wasserstein distances, we set the costs to be infinity. We summarize the pseudo-code in algorithm 3, where SOLVEOT represents any OT solver, e.g. (Cuturi, 2013), and UPDATEPLAN is the subroutine in Algorithm 1. The side-product $\mathbf{P}_{in}$ is a tensor that allows us to do accurate point to point alignment. Based on it, we propose the following estimated transportation of $\mathbf{x}_i$,

$$\hat{x}_i = \mathbf{P}_{in}[m^*, n^*, i, :]\mathbf{Y}, \text{ where } m^* = \operatorname*{argmax}_m \mathbf{P}_x[i,m], \text{and } n^* = \operatorname*{argmax}_n \mathbf{P}_z[m^*, n]. \tag{43}$$

C. Variants of LOT

In this section, we discuss some natural extensions of LOT and their applications.

1) Data alignment under a global transformation: In many alignment applications, it is often the case that the learned features of the source and target are subject to some shift or transformation (Alvarez-Melis et al., 2019). For example, in neuroscience (Lee et al., 2019), subspaces of the data clusters from neural and motor activities are subject to transformations preserving their principal angles. Formally, this is equivalent to saying that a transformation from the Stiefel manifold $\mathcal{V}_d := \{\mathbf{O} \in \mathbb{R}^{d \times d} : \mathbf{O}^T\mathbf{O} = \mathbf{I}_d\}$ exists between them. Using the squared ℓ_2-distance as the ground truth metric, we can thus pose an OT problem under this transformation as,

$$\min_{\mathbf{M} \in \mathcal{V}_d} \min_{\mathbf{P}\mathbb{1} = \mu, \mathbf{P}^T\mathbb{1} = \nu} \sum_{i=1}^N \sum_{j=1}^M \|\mathbf{M}\mathbf{x}_i - \mathbf{y}_j\|_2^2 \mathbf{P}_{i,j}, \tag{44}$$

Algorithm 4 Unbalanced Latent Optimal Transport

Input: $\mathbf{K}_x, \mathbf{K}_y, \mathbf{K}_z, \tau_1, \tau_2, \epsilon$.

Output: $\mathbf{P}_x, \mathbf{P}_y, \mathbf{P}_z, \mathbf{u}_z, \mathbf{v}_z, \mathbf{z}_1, \mathbf{z}_2$

Initialize: $\alpha_x \leftarrow \mathbb{1}_N, \beta_x \leftarrow \mathbb{1}_{k_1}, \alpha_y \leftarrow \mathbb{1}_{k_2}, \beta_y \leftarrow \mathbb{1}_M, \alpha_z \leftarrow \mathbb{1}_{k_1}, \beta_z \leftarrow \mathbb{1}_{k_2}, \mathbf{z}_1 \leftarrow \mu, \mathbf{z}_2 \leftarrow \nu$

1: **while** not converging **do**

2: $(\mathbf{z}_1, \alpha_x) \leftarrow \left((\alpha_x \odot \mathbf{K}_x \beta_x)^{\frac{\epsilon}{\tau_1+\epsilon}} \odot \mathbf{z}_1^{\frac{\tau_1}{\tau_1+\epsilon}}, \quad \alpha_x^{\frac{\epsilon}{\tau_1+\epsilon}} \odot (\mathbf{z}_1 \oslash \mathbf{K}_x \beta_x)^{\frac{\tau_1}{\tau_1+\epsilon}} \right)$

3: $(\mathbf{z}_2, \beta_y) \leftarrow \left((\beta_y \odot \mathbf{K}_y^T \alpha_y)^{\frac{\epsilon}{\tau_2+\epsilon}} \odot \mathbf{z}_2^{\frac{\tau_2}{\tau_2+\epsilon}}, \quad \beta_y^{\frac{\epsilon}{\tau_2+\epsilon}} \odot (\mathbf{z}_2 \oslash \mathbf{K}_y^T \alpha_x)^{\frac{\tau_2}{\tau_2+\epsilon}} \right)$

4: $\mathbf{u}_z \leftarrow ((\alpha_z \odot \mathbf{K}_z \beta_z) \odot (\beta_x \odot \mathbf{K}_x^T \alpha_x))^{\frac{1}{2}}$

5: $\beta_x \leftarrow \mathbf{u}_z \oslash \mathbf{K}_x^T \alpha_x$; $\alpha_z \leftarrow \mathbf{u}_z \oslash \mathbf{K}_z \beta_z$

6: $\mathbf{v}_z \leftarrow ((\alpha_y \odot \mathbf{K}_y \beta_y) \odot (\beta_z \odot \mathbf{K}_z^T \alpha_z))^{\frac{1}{2}}$

7: $\beta_z \leftarrow w_{zy} \oslash \mathbf{K}_z^T \alpha_z$; $\alpha_y \leftarrow \mathbf{v}_z \oslash K_y \beta_y$

8: **end while**

Return: $\mathbf{P}_x = \mathbf{D}(\alpha_x)\mathbf{K}_x\mathbf{D}(\beta_x), \mathbf{P}_y = \mathbf{D}(\alpha_y)\mathbf{K}_y\mathbf{D}(\beta_y), \mathbf{P}_z = \mathbf{D}(\alpha_z)\mathbf{K}_z\mathbf{D}(\beta_z), \mathbf{u}_z, \mathbf{v}_z, \mathbf{z}_1, \mathbf{z}_2$

where d is the dimension of the data. Observe that (44) is a sum over the number of all data points (with an order of $\mathcal{O}(NM)$), it could be fragile to outliers or noise. On the other hand, LOT could remedy the issue by casting an optimization only on the anchors $\mathbf{z}^x$, $\mathbf{z}^y$. Since the anchors are some averaged representations of data points, they are smooth and robust to data perturbations. The resultant optimization is,

$$\min_{\mathbf{M}\in\mathcal{V}_d} \min_{\mathbf{P}_z\mathbb{1}=\mu_z, \mathbf{P}_z{}^T\mathbb{1}=\nu_z, \mathbf{P}_x, \mathbf{P}_y} \langle \mathbf{C}_x, \mathbf{P}_x \rangle + \sum_{i=1}^{k_1}\sum_{j=1}^{k_2} \|\mathbf{M}\mathbf{z}_i^x - \mathbf{z}_j^y\|_2^2 (\mathbf{P}_z)_{i,j} + \langle \mathbf{C}_y, \mathbf{P}_y \rangle. \tag{45}$$

Now the order of the number of terms in the objective is greatly reduced to $\mathcal{O}(k_1 k_2)$. As the transformation is cast upon the anchors, the solution tends to be more robust to perturbations.

Below we propose a simple procedure to optimize $\mathbf{M}$. Our method is based on an alternating strategy, where we alternate between two procedures: (1) $\mathbf{P}_x, \mathbf{P}_z, \mathbf{P}_y$ are fixed when we update $\mathbf{M}$; (2) $\mathbf{M}$ is fixed when we update $\mathbf{P}_x, \mathbf{P}_z, \mathbf{P}_y$. The later is nothing more than the algorithm 1 with $\mathbf{C}_z := [\|\mathbf{M}\mathbf{z}_i^x - \mathbf{z}_j^y\|_2^2]_{i,j}$. Hence we would focus on the procedure (1). Given a fixed $\mathbf{P}_z^*$, the minimization of the middle term of (45), after simplification, is equivalent to,

$$\max_{\mathbf{M}\in\mathcal{V}_d} \langle \mathbf{M}\mathbf{Z}_x\mathbf{P}_z^*, \mathbf{Z}_y \rangle. \tag{46}$$

The problem admits an elegant solution according to the following lemma.

Lemma 3. *(Lemma 3.1 (Alvarez-Melis et al., 2019)) Let* $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ *be the SVD decompostion of* $\mathbf{Z}_y(\mathbf{Z}_x\mathbf{P}_z^*)^T$*, then* $\mathbf{M}^* = \mathbf{U}\mathbf{V}^T$ *is the optimal solution to (46).*

Finally, we note that the class of transformations depends on the application, and the applicability of LOT could go beyond the Stiefel manifold considered here.

2) Unbalanced variant of LOT: There are often cases in practice where the distribution of data is not normalized so the total mass of the source and target are not equal. In this case, OT preserving marginal distributions does not admit a solution. To deal with this problem, the unbalanced optimal transport (UOT) is among one of the most prominent proposed method (Liero et al., 2018). Instead of putting hard constraints on the marginal distributions, UOT considers an optimization problem regularized by the deviation to the marginal distributions measured by the KL-divergence. Just as in OT, LOT can be naturally extended in this case by modifying the objective in Eqn. (8) to,

$$\begin{aligned} &\min_{\mathbf{P}_x, \mathbf{P}_y, \mathbf{P}_z, \mathbf{z}_1, \mathbf{z}_2} \epsilon \mathrm{KL}(\mathbf{P}_x|\mathbf{K}_x) + \tau_1 \mathrm{KL}(\mathbf{z}_1|\mu) + \epsilon \mathrm{KL}(\mathbf{P}_z|\mathbf{K}_z) + \epsilon \mathrm{KL}(\mathbf{P}_y|\mathbf{K}_y) + \tau_2 \mathrm{KL}(\mathbf{z}_2|\nu) \\ &\text{subject to: } \mathbf{P}_x\mathbb{1} = \mathbf{z}_1, \mathbf{P}_x^T\mathbb{1} = \mathbf{u}_z, \mathbf{P}_z\mathbb{1} = \mathbf{u}_z, \mathbf{P}_z^T\mathbb{1} = \mathbf{v}_z, \mathbf{P}_y\mathbb{1} = \mathbf{v}_z, \mathbf{P}_y^T\mathbb{1} = \mathbf{z}_2. \end{aligned} \tag{47}$$

The algorithm could be derived similarly as in Appendix B.1, and we provide the pseudocode in Algorithm 4.

3) Barycenter of hubs: Finding the barycenter is a classical problem of summarizing several data points with a single "mean". In optimal transport, this amounts to the Wasserstein barycenter problem (Cuturi & Doucet, 2014). LOT casts an unique variant of the problem in the following scenario. Consider a shipping company ships items worldwide. The company has several hubs in its country to gather the items and then ship them to other countries. On the other hand, each country also has its own receiving hubs to receive the items. The company then must decide the locations of its hubs in its country to minimize the transport effort. The above scenario poses an unique barycenter problem in terms of optimizing locations of the hubs. By using an analogy of hubs to anchors, we can transform the problem into the framework of LOT , where one seeks the optimal sources' anchors Z_x by solving the optimization,

$$\min_{Z_x} \min_{Z_{y_i}, i\in[K]} \sum_{k=1}^{K} \mathrm{OT}^{\mathrm{L}}(\mu, \nu_k). \tag{48}$$

Note the inner optimization is related to the hubs of the receiving countries, which depending on the applications, could be either fixed or free to optimize. To solve this problem, one can simply extend Algorithm 1 to make it distributed. This is done by applying Algorithm 1 to each term in the summation.

D. Additional experiments

In this section, we describe the results for additional experiments that carried out to test LOT .

D.1. Domain adaptation task

Neural network implementation details: The classifier is a multi-layer perceptron with two 256-units hidden layers and ReLU activation functions. We use dropout to regularize the network. 28×28 Images are first normalized using mean $= 0.1307$ and std $= 0.3081$, then flattened to 784 dimensions before being fed to the network, which then outputs a 10-d logits vector. The classifier is trained using a cross-entropy loss. To get the predicted class, we used the argmax operator. The network only learns from the training samples of MNIST, and then its weights are frozen. USPS images are smaller with 16×16 pixels. To feed them to our classifier, we apply zero-padding to get the desired input size.

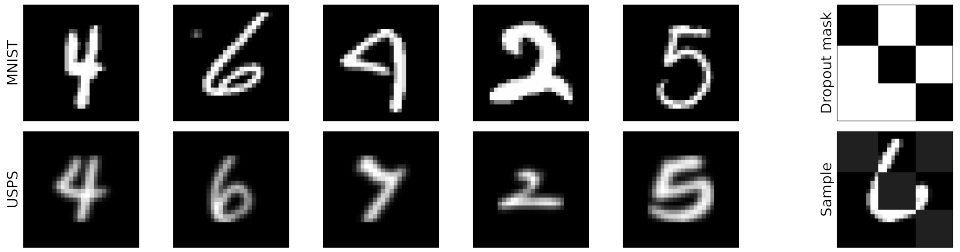
Domain shift in neural networks experiments: For our experiment, we use 1000 randomly sampled images from each of these sets: the training and testing sets from MNIST and the testing set from USPS. The sampling isn't done in any particular way, so there is a different number of samples for each class in each set. For all experiments using the classifier, we use the argmax operator on the transported features to get the predictions after aligning source and target. We then compare to the ground truth labels to get the accuracy. All results can be found in Table S1.

For the first experiment (Table S1-DA), we align the testing set of USPS with the training set of MNIST (Figure S1a). For the second and third experiments (Table S1-D, DU), we study the influence of perturbations on the output space of the classifier. So we align a perturbed version of the testing set of MNIST with the training set of MNIST. For both experiments, we use coarse dropout, which sets rectangular areas within images to zero. These areas have large sizes. We generate a coarse dropout mask once and apply it to all images in our testing set (Figure S1b). In addition to coarse dropout, we also eliminate some classes (2, 4, and 8 digit classes) to get the unbalanced case for the third experiment (Table S1-DU).

Visualization of transported distributions: In Figure 4a, we project the distribution of the neural network's output features (for each set) in 2D using Isomap fit to the MNIST training set's output features. We use 50 neighbor points in the Isomap algorithm to get a reasonable estimate of the geodesic distance, as we have 10 well-separated clusters in the output space of the deep neural network in the case of MNIST's training set. To ensure consistency, we use the projection learned on the MNIST training set with all other sets.

Visualization of transport plans: We provide further visualization of the transport plans obtained by LOT and FC , for the Domain Adaptation experiment and the unbalanced alignment experiment (Figure S2).

Visualization of transported points: To get a better understanding of how the embedding of each sample is changing after transportation, we provide visualizations of where samples are being transported to in the target space. To describe the target space locally, we find the five nearest neighbors of a sample according to the L2 distance between the transported



(a) MNIST-USPS Samples (b) Dropout Mask

Figure S1: *Samples used in deep neural network experiments.* (a) MNIST samples vs. USPS samples (after zero-padding), used in the Domain Adaptation experiment. (b) Coarse dropout mask applied over all samples in MNIST's testing set and a sample of the perturbed images, used in the domain shift experiment.

	MNIST-USPS (DA)	MNIST (D)		MNIST (DU)	
	Accuracy	Accuracy	L2 error	Accuracy	L2 error
Original	79.3	65.7	0.74	72.6	0.72
OT	76.9	73.4	0.59	61.5	0.71
kOT	79.4	74.0	0.53	60.9	0.73
SA	81.3	64.9	-	72.3	-
FC	84.1	77.6	**0.51**	67.2	0.59
LOT-L2	**86.2**	**78.2**	0.53	**77.7**	**0.56**

Table S1: ***Results for concept drift and domain adaptation for handwritten digits.*** The classification accuracy and L2-error (transported samples vs. ground truth test samples) are computed for the synthetic drift experiment: Coarse dropout (left) and for the domain adaptation experiment: MNIST to USPS (right). Our method is compared with the accuracy before alignment (Original), entropy-regularized OT, k-means plus OT (kOT), and subspace alignment (SA).

features of the source sample and the features of the target samples. In Figure S3, we show some of the cases where the three methods, OT , FC and LOT , disagree.

In the top two rows (Figure S3a-b), we see cases where LOT outperforms OT and FC . When all neighbors have the same labels, we can safely assume that we are not in a boundary between classes but deep within a class cluster, so this reflects on the confidence these methods have in their transportation. Both OT and FC confidently map the sample to the wrong region of the space (Figure S3a).

In the bottom two rows (Figure S3c-d), we see more difficult cases. In (Figure S3c), we see that even though the closest neighbor (4 in light green) doesn't correspond to the ground truth label (7), LOT-L2 seems to be mapping the point to the boundary between classes 7 and 4, and it does in fact classify it correctly. Lastly, in (d), we see that none of the methods is able to recover information lost due to dropout.

D.2. Additional synthetic experiments

We study the behaviors of LOT for two synthetic experiments performed in (Forrow et al., 2019; Paty & Cuturi, 2019).

Fragmented Hypercube: In the first experiment, the source distribution is $\mu = \mathcal{U}([-1,1]^d)$ and the target distribution is $\nu = T_{\#}\mu$ (i.e., $\nu \overset{d}{=} T(x), x \overset{d}{=} \mu$), where $T(x) = x + 2\text{sign}(x) \odot (e_1 + e_2)$, and e_i is the canonical basis of $\mathbf{R}^d$. In this example, the signal lies only in the first 2 dimensions while the remaining $d-2$ dimensions are noises. The data shows explicit clustering structure. We investigate the estimated transports $\hat{x} = \int yp(y|x)dx$ produced by OT, entropy- regularized OT, and LOT with $k_x = k_y = 4$ to see if LOT can capture the data structure and whether it provides robust transport. We choose $d = 30$ and draw $N = 250$ points from each distribution. The result is shown in Figure S4. We see that all methods capture the cluster structure, but both OT and regularized are sensitive to noise in $d-2$ dimensions, while LOT provides a data transport that is more robust against noise in high dimensional space.

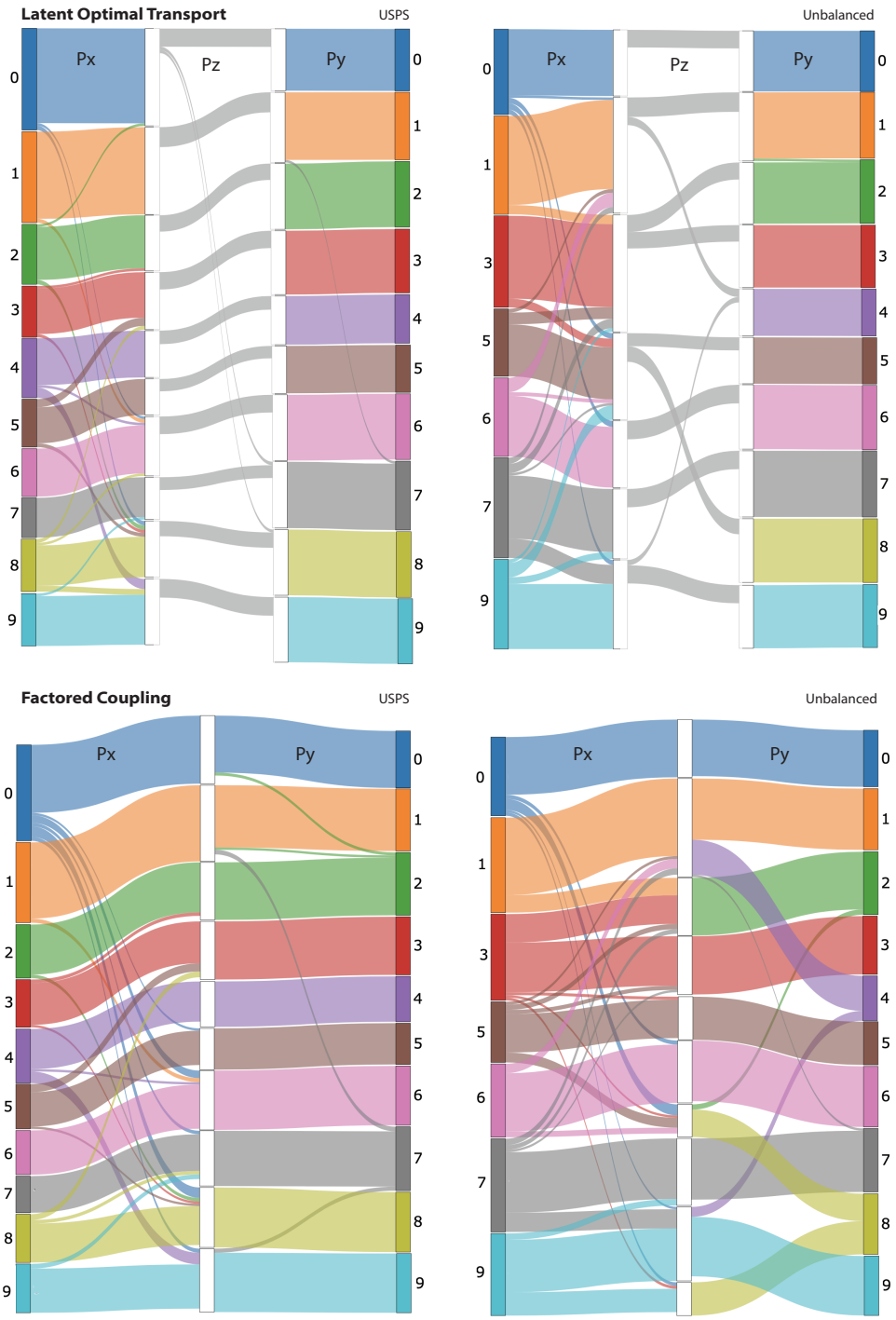


Figure S2: ***Visualization of transport plans obtained with `LOT` and factored coupling.*** We show the transport plans for `LOT` (top) and `FC` (bottom) for the USPS (left) and unbalanced dropout example (right).

Disk to annulus: In the second experiment, we consider data without separate clustering structure. In this example, the source and target distributions are

$$\mu = \mathcal{U}\{\mathbf{x} \in \mathbb{R}^d : 0 \leq \|(\mathbf{x}_1, \mathbf{x}_2)\|_2 \leq 1, \mathbf{x}_i \in \{0, 1\}, i = 3, \ldots, d\}$$
$$\nu = \mathcal{U}\{\mathbf{x} \in \mathbb{R}^d : 2 \leq \|(\mathbf{x}_1, \mathbf{x}_2)\|_2 \leq 3, \mathbf{x}_i \in \{0, 1\}, i = 3, \ldots, d\}.$$

Again only the first 2 dimensions contain the signal while the rest $d-2$ are noise. We draw 250 points from each distribution and choose $d = 30$. We compare the estimated transports $\hat{x} = \int yp(y|x)dx$ for OT, entropy-regularized OT, and LOT with $k_x = k_y = 15$. In this example, we increase the number of anchors as the annulus can be regarded as having infinite clusters. We visualize the result in Figure S5. We observe that LOT is more regular than the other two, but the support of the target is not fully covered by the transportation. This is because the rank constraint imposed on LOT limits its degree of freedom to transport data.

D.3. Examining the impact of cluster correlation on performance

Next, we study the ability of LOT of capturing clustering structure as the correlation between clusters increases. It is observed in (Lee et al., 2019) that equally spaced clusters (i.e., uniformly distributed subspace angles) are harder to align. In the spirit of measuring cluster similarity as in (Soltanolkotabi et al., 2012), we define the cross-correlation between two clusters $\mathcal{Y}_i$ and $\mathcal{Y}_j$ as $\mu_c(\mathcal{Y}_i, \mathcal{Y}_j) := \sum_{x\in\mathcal{Y}_i, y\in\mathcal{Y}_i} \langle x, y\rangle / |\mathcal{Y}_i||\mathcal{Y}_j|$. In Fig. S6, we plot the transport plans for GMM with 5 clusters in two different correlation regimes: (i) the top row shows the regime where the cross-correlation is strong within the same class; (ii) the bottom row shows the uniform correlation case where clusters are approximately equally spaced and the correlations are much more uniform across classes. We observe in both results that LOT has transport plans similar to the cross-correlation in the pattern, while for uniform case, the transport plan of OT seems less structured. This example suggests that LOT can be a useful tool for data visualization.

E. Details of GMM Experiment (E2)

In this experiment, we use a Gaussian mixture model (GMM) with M components in $\mathbb{R}^d$ to generate the data for the source and target. For each component, the mean is randomly generated from a random standard normal distribution, and the covariance matrix is generated from a Wishart distribution. To model low-dimensional latent structure, we then apply random k-dimensional projection and add a d-dimensional standard normal noise to each component. The source and target draw 100 points from each component independently.

Below we provide the details to each plot of (a) to (e). The default k is set to 5.

Parameters of the data generation:

- (a) We set $M = 4, d = 30$. We choose a random unit vector and rotate the mean of the component by θ-degree using the vector as the axis.
- (b) $M = 4, d = 30$. For any ratio r of outliers, we randomly pick up rn number of points and modify its distribution by a linear combination of Gaussian noise and the original point. The variance of Gaussian noise is equal to half the squared mean of the component.
- (c) $M = 4$. We increase the dimension d. As the signal lies in a k-dimensional space, the rest of $d - k$-dimensions are noisy.
- (d) $d = 30$. Here we generate a GMM with 10 components. While the target has 10 classes with 100 points per class initially, we vary the number of the source class from 2 to 10. The x-axis denotes the ratio between the number of components to the source and to the target.
- (e) $d = 30, M = 4$. We set the number of the source and target anchors to be equal and increase the number from 2 to 100. Note that this number also upper bounds the rank of the associated transport plan.

Algorithmic implementation:

- We use the POT: Python Optimal Transport package (Flamary & Courty, 2017) https://pythonot.github.io/ as the implementation for a vanilla optimal transport.
- In all of our GMM experiments, the entropy regularization parameter ε is set to 10.

- For the transport estimation, we use the expected transportation defined by a transport plan, which is:

$$\hat{\mathbf{X}} = \text{diag}(\mu)\mathbf{P}\mathbf{Z}_2, \text{ for OT .} \tag{49}$$

$$\hat{\mathbf{X}} = \text{diag}(\mu)\mathbf{P}_x(\mathbf{Q}_y - \mathbf{Q}_x), \text{ for FC (Forrow et al., 2019)} \tag{50}$$

$$\hat{\mathbf{X}} = \text{diag}(\mu^{-1})\mathbf{P}_x\text{diag}((\mathbf{P}_z\mathbf{1})^{-1})\mathbf{P}_z(\mathbf{Q}_y - \mathbf{Q}_x), \text{ for LOT (the estimator in Section 3.4),} \tag{51}$$

  where $\mathbf{Q}_x = \text{diag}(\mu^{-1})\mathbf{P}_x^T\mathbf{X}^T$, $\mathbf{Q}_y = \text{diag}(\nu^{-1})\mathbf{P}_y\mathbf{Y}$ denote the centroids for FC and LOT .

- We adopt the 1-NN classification rule where the class of a point from the source is predicted by the class of the nearest neighbor (among the target points) of its estimated transportation.

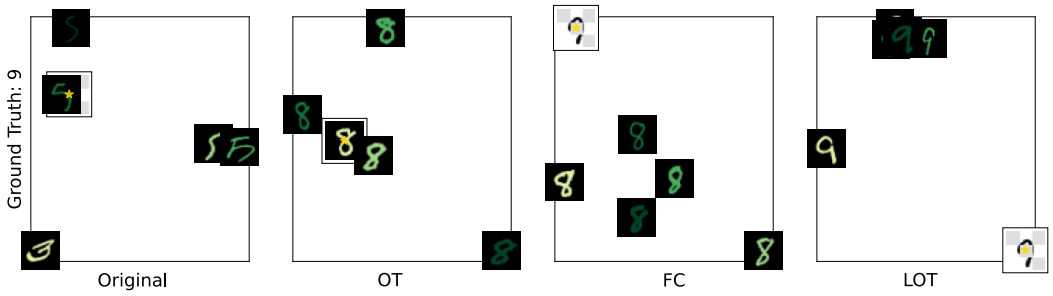
F. Choice of hyperparameters

Code availability: We provide an implementation of LOT and a demo for a dropout experiment on MNIST in the supplementary file. We use the POT: Python Optimal Transport package (Flamary & Courty, 2017) https://pythonot.github.io/ as the implementation for a vanilla optimal transport.

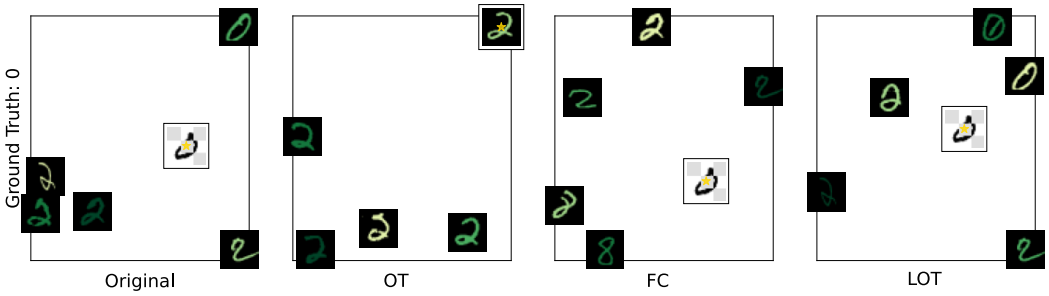
Hyperparameter tuning: LOT has two main hyperparameters that must be specified: (i) the number of anchors and (ii) the entropy-regularization parameter epsilon. For domain adaptation experiments (E2-3), the number of anchors was set to 10, using the prior we had about there being 10 classes of digits. The regularization parameter epsilon was set to 50; this parameter depends on the scale of the data. We note that in the case of (E2), the standard deviation of the logit outputs of the network was around 10^3. More generally, we proceed as follows when choosing the hyperparameters:

- **Number of anchors.** To optimize the number of anchors, one can use domain knowledge or model selection procedures common in clustering. For instance, in the case of MNIST-USPS, we know to expect 10 clusters, each corresponding to a digit, and set the number of anchors to be at least this amount. In cases where we do not have a priori knowledge about the number of clusters in the data, we can use standard approaches for model selection in clustering (e.g., silhouette score (Rousseeuw, 1987), Calinski-Harabaz index (Caliński & Harabasz, 1974)). In our experiments, we find that the number of anchors can be increased progressively before observing a phase transition (or change point) where the accuracy increases significantly (see Figure 2.e). This change point typically coincides with the number of clusters in the data and can be used to select the number of anchors. Furthermore, while some interpretability may be sacrificed when we overestimate the number of anchors, we find that this does not hinder the performance of the method in terms of the quality of overall transport (see Figure 2.e). This is reminiscent of the elbow phenomena found with clustering methods, so we can imagine utilizing established clustering metrics for tuning these two parameters. This is left for future work, as the simple elbow method was found to be sufficient.

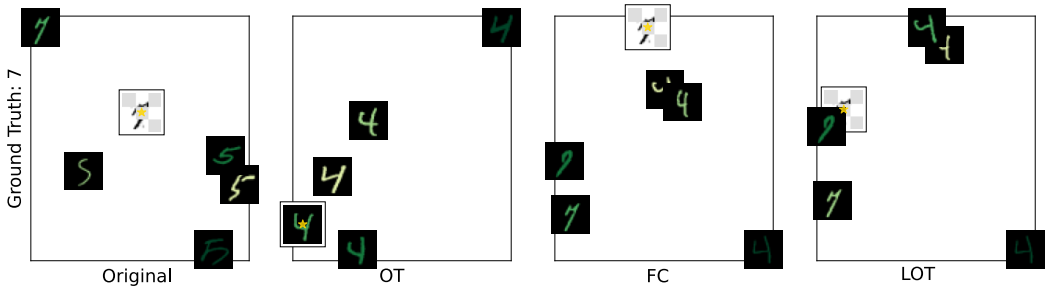
- **Entropy- regularization parameter.** The choice of the regularization parameter mainly depends on the scale of the data. In our experiments, we search for the lowest epsilon for which LOT converges using a simple Bisection method. This same approach is used for selecting epsilon for OT in our experiments. We find that the regularization value is inherent to the data and not the task (Exp 2).



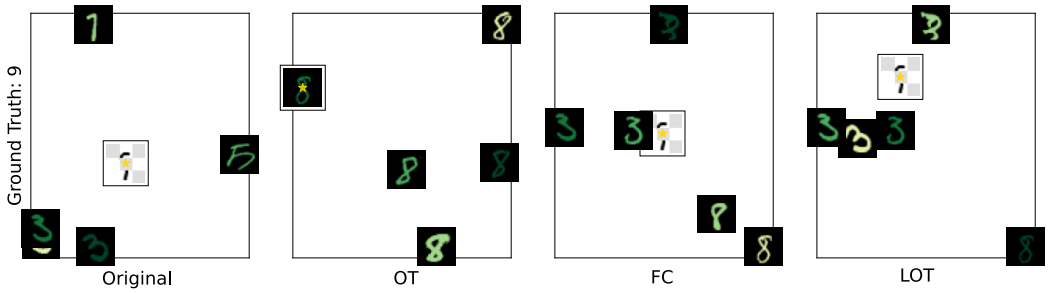
(a)



(b)



(c)



(d)

Figure S3: ***Neighborhood change after transportation (DU)*** For the unbalanced transport experiment, we show for multiple perturbed samples (white, marked with a $\star$) the change in the closest neighbors landscape before (Original) and after the transportation, for different methods (OT , `FC` and `LOT`). We find the five nearest neighbors (black) according to the L2 distance between the transported output features of the sample and the features from the target training samples, and arrange the samples in 2D space using the Isomap projection. We apply a gradient filter over the neighbors' images such that the closest neighbor is light green.

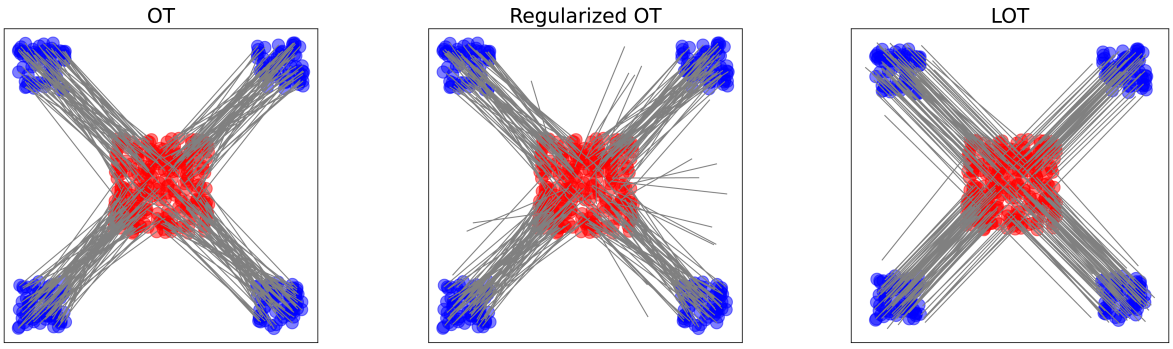


Figure S4: ***Fragmented Hypercube.*** We visualize the estimated transports $\hat{x} = \int yp(y|x)dx$ for OT, regularized OT, and `LOT` .

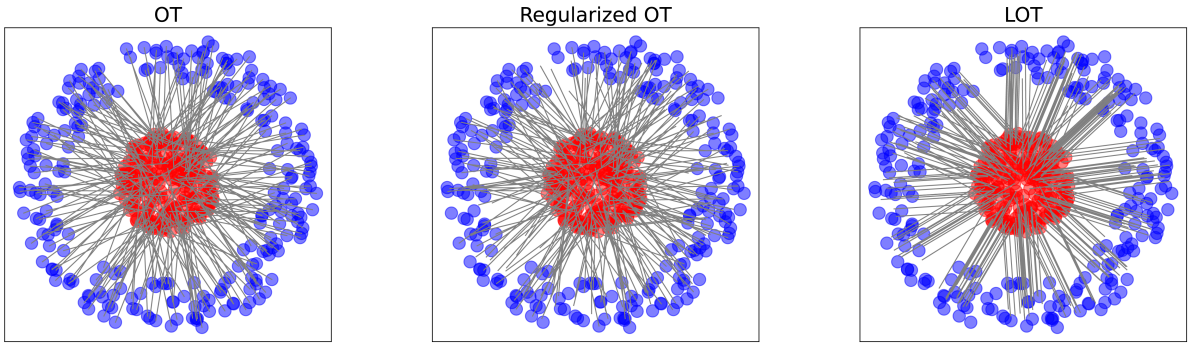
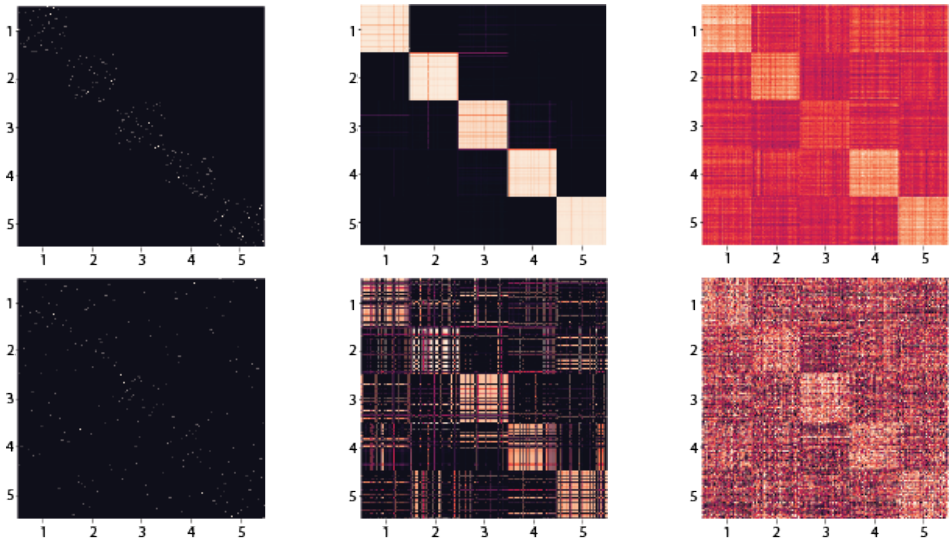


Figure S5: ***Disk to Annulus.*** We visualize the estimated transports $\hat{x} = \int yp(y|x)dx$ for OT, regularized OT, and `LOT`.



(a) Optimal Transport (b) `LOT-L2` (c) Cross-Correlation

Figure S6: **Examining the connection between cluster coherence and transport.** In (a-b), we visualize the transport plan obtained by OT and `LOT` when the different mixture components in the model are weakly correlated (top) and strongly correlated (bottom). In (c), we examine the cross-correlation between points in the source and target for the two conditions.