Appendix

In the appendix, we provide the proof of Theorem 1, the optimization on PUOT, SSLM and more experiential results.

A. Proof of Theorem 1

Theorem 1. *Given the PUOT problem J_{puot} as shown in Eq.(2), it has the following dual form:*

$$\min_{\boldsymbol{f},\boldsymbol{g},\zeta} \left[\tau \sum_{i=1}^{N} a_i \exp\left(-\frac{f_i+\zeta}{\tau}\right) - \sum_{j=1}^{H} b_j (g_j - \zeta) \right] \quad (13)$$
$$s.t.\ f_i + g_j + s_{ij} = \mathcal{U}_{ij}, \quad s_{ij} \geq 0,$$

where $\boldsymbol{f}$, $\boldsymbol{g}$, $\boldsymbol{s}$ and ζ denote Lagrange multipliers. Meanwhile $s_{ij} \geq 0$ according to the KKT conditions. Moreover, PUOT will be transformed into sample-reweighted optimal transport problem according to its dual form:

$$\min_{\boldsymbol{\pi}\geq 0} \langle \boldsymbol{\mathcal{U}}, \boldsymbol{\pi} \rangle$$
$$s.t. \begin{cases} \sum_{j=1}^{H} \pi_{ij} = a_i \exp\left(-\frac{f_i^* + \zeta^*}{\tau}\right) \\ \sum_{i=1}^{N} \pi_{ij} = b_j \end{cases} \quad (14)$$

Proof. Firstly, we suppose the source marginal probability is defined as $\boldsymbol{\alpha}$ and we have $\boldsymbol{\pi}\mathbf{1}_H = \boldsymbol{\alpha}$. Note that we do not need to know the exact value of $\boldsymbol{\alpha}$ beforehand. We adopt this optional constraint only for simplifying the following deduction. Meanwhile the PUOT problem J_{puot} can be rewritten as below:

$$\min_{\pi_{ij}\geq 0} J_{\text{puot}} = \langle \boldsymbol{\mathcal{U}}, \boldsymbol{\pi} \rangle + \tau \text{KL}\left(\boldsymbol{\pi}\mathbf{1}_H \| \boldsymbol{a}\right)$$
$$\text{s.t.} \begin{cases} (\text{Constraint}): \ \boldsymbol{\pi}^\top \mathbf{1}_N = \boldsymbol{b} \\ (\text{Optional}): \ \boldsymbol{\pi}\mathbf{1}_H = \boldsymbol{\alpha} \end{cases} \quad (15)$$

Then the Lagrange multipliers of PUOT can be shown as:

$$\max_{\boldsymbol{s}\geq 0, \boldsymbol{f},\boldsymbol{g},\zeta} \min_{\boldsymbol{\pi}\geq 0} \mathcal{J} = \tau \text{KL}\left(\boldsymbol{\pi}\mathbf{1}_H \| \boldsymbol{a}\right) + \langle \boldsymbol{f} + \zeta, \boldsymbol{\pi}\mathbf{1}_H \rangle + \langle \boldsymbol{g} - \zeta, \boldsymbol{b} \rangle + \mathscr{C}_{\text{SUOT}} \quad (16)$$

where $\boldsymbol{f}$, $\boldsymbol{g}$, ζ and $\boldsymbol{s}$ denote Lagrange multipliers. Note that ζ can also be viewed as the translation invariant term [79]. Meanwhile $\mathscr{C}_{\text{SUOT}}$ can be calculated as:

$$\begin{aligned} \mathscr{C}_{\text{SUOT}} &= \sum_{i,j} (\mathcal{U}_{ij} - f_i - g_j - s_{ij}) \pi_{ij} \\ &= \langle \boldsymbol{\mathcal{U}} - \boldsymbol{f} \otimes \mathbf{1}_N^\top - \mathbf{1}_H \otimes \boldsymbol{g}^\top - \boldsymbol{s}, \boldsymbol{\pi} \rangle \\ &= \langle \boldsymbol{\mathcal{U}} - \boldsymbol{f} \otimes \mathbf{1}_N^\top - \mathbf{1}_H \otimes \boldsymbol{g}^\top, \boldsymbol{\pi} \rangle \end{aligned} \quad (17)$$

where $s_{ij} \geq 0$ and $s_{ij}\pi_{ij} = 0$ according to the KKT conditions. Reviewing the definition of the KL-Divergence is given as $\text{KL}\left(\boldsymbol{x}\|\boldsymbol{z}\right) = \sum_{d=1}^{D} [x_d \log(x_d/z_d) - x_d + z_d]$, we have the following the result:

$$\frac{\partial}{\partial \pi_{ij}} \left(\text{KL}\left(\boldsymbol{\pi}\mathbf{1}_H \| \boldsymbol{a}\right)\right) = \log \frac{\sum_{j=1}^{H} \pi_{ij}}{a_i} \quad (18)$$

By further taking the differentiation on π_{ij} we have:

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \pi_{ij}} &= \left[\tau \log \frac{\sum_{j=1}^{H} \pi_{ij}}{a_i} + f_i + \zeta \right] + (\mathcal{U}_{ij} - f_i - g_j - s_{ij}) \\ &= \mathcal{U}_{ij} + \tau \log \frac{\sum_{j=1}^{H} \pi_{ij}}{a_i} + \zeta - g_j - s_{ij} = 0 \end{aligned} \quad (19)$$

Here we can reach the following conclusions:

$$\begin{cases} \sum_{j=1}^{H} \pi_{ij} = a_i \exp\left(-\frac{f_i+\zeta}{\tau}\right) \\ \sum_{i=1}^{M} \pi_{ij} = b_j \end{cases} \quad (20)$$

We can easily verify that the above equations satisfies the condition $\mathcal{U}_{ij} - f_i - g_j - s_{ij} = 0$. Once we take the marginal probabilities into KL-Divergence, we will obtain the results:

$$\begin{aligned} \mathscr{L} &= \tau \text{KL}\left(\boldsymbol{\pi}\mathbf{1}_H \| \boldsymbol{a}\right) + \langle \boldsymbol{f} + \zeta, \boldsymbol{\pi}\mathbf{1}_H \rangle \\ &= \tau \text{KL}\left(\left\langle \boldsymbol{a}, \exp\left(-\frac{\boldsymbol{f}+\zeta}{\tau}\right)\right\rangle \| \boldsymbol{a}\right) + \left\langle \boldsymbol{f} + \zeta, \boldsymbol{a} \exp\left(-\frac{\boldsymbol{f}+\zeta}{\tau}\right)\right\rangle \\ &= \sum_{i=1}^{N} \left[-\tau a_i \exp\left(-\frac{f_i+\zeta}{\tau}\right) + \tau a_i \right] \end{aligned} \quad (21)$$

By neglecting the irrelevant constants, we can reach out the final solution:

$$\max_{\boldsymbol{f},\boldsymbol{g},\zeta} \left[-\tau \sum_{i=1}^{N} a_i \exp\left(-\frac{f_i+\zeta}{\tau}\right) + \sum_{j=1}^{H} b_j (g_j - \zeta) \right]$$
$$s.t.\ f_i + g_j + s_{ij} = \mathcal{U}_{ij}, \quad s_{ij} \geq 0, \quad (22)$$

It is obvious to verify the equivalence between Eq.(22) and Eq.(13) and we finish the proof of Theorem 1. □

B. Optimization on PUOT

In this section, we will provide the optimization details on solving PUOT via dealing with variables $\boldsymbol{f}$ and ζ in the following equation:

$$\min_{\boldsymbol{f},\zeta} \mathcal{L}_{\text{F}} = \tau \sum_{i=1}^{N} a_i \exp\left(-\frac{f_i+\zeta}{\tau}\right) - \sum_{j=1}^{H} \left(\inf_{k\in[N]} (\mathcal{U}_{kj} - f_k) - \zeta \right) b_j. \quad (23)$$

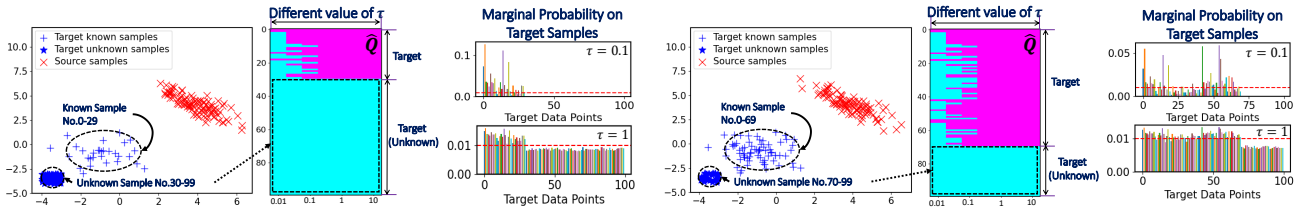

Figure 8. The illustration on the proposed weight barcode estimation with toy examples. We first sample 100 source/target data samples and set 30% or 70% of target data samples from unknown categories. Then we adopt marginal probability estimation to obtain the results of target weights. Finally we calculate and visualize the result of $\widehat{\boldsymbol{Q}}$ to distinguish the unknown target samples.

Algorithm 2 The optimization procedure on PUOT

Input: $\boldsymbol{\mathcal{U}}$: The cost distance matrix; τ: Hyper parameters; $\boldsymbol{a}, \boldsymbol{b}$: Given the source and target marginal probabilities.

Procedure:

1: Initialize $t = 0$, $\zeta = 0$, $\boldsymbol{f}^{(0)} = (0, 0, \cdots, 0)$.
2: **repeat**
3: Adopt WFPI to optimize $\boldsymbol{f}$ as:

$$f_s^{(\ell+1)} = \kappa\varsigma \left[\log(a_s) - \frac{\zeta}{\tau} - \log\left(\sum_{j=1}^{H} \frac{b_j e^{-\mathcal{U}_{sj}/\varsigma}}{\mathscr{M}_j} \right) \right] = \mathscr{U}_s\left(f_1^{(\ell)}, \cdots, f_s^{(\ell)}, \cdots, f_N^{(\ell)}\right)$$

4: Update ζ using $\boldsymbol{f}$ via Eq.(29).
5: Update $t = t + 1$.
6: **until** Converge
7: **Return**: The optimal ζ^* and $\boldsymbol{f}^*$.

To further accelerate the optimization process, we consider to make a smooth approximation on replacing $\inf(\cdot)$ as $\inf_{k\in[N]}[\mathcal{U}_{kj} - f_k] \approx -\varsigma \log[\sum_{k=1}^{N} e^{\frac{f_k - \mathcal{U}_{kj}}{\varsigma}}]$. Note that $\varsigma > 0$ denotes the balanced hyper parameters among the accuracy and function smoothness. In the experiment, we set $\varsigma = 0.01$ for the calculation. Then we can obtain the following optimization problem:

$$\min_{\boldsymbol{f},\zeta} \mathscr{A}_{\mathrm{PUOT}} = \tau \exp\left(-\frac{\zeta}{\tau}\right) \sum_{i=1}^{N} a_i \exp\left(-\frac{f_i}{\tau}\right) + \sum_{j=1}^{H} \left[\varsigma \log\left[\sum_{k=1}^{N} \exp\left(\frac{f_k - \mathcal{U}_{kj}}{\varsigma} \right) \right] + \zeta \right] b_j \tag{24}$$

Therefore, we further propose Wasserstein Fixed-Point Iteration (WFPI) method to optimize PUOT. That is, we take the differentiation with respect to f_i and set $\partial \mathscr{A}_{\mathrm{PUOT}} / \partial f_i = 0$as:

$$\exp\left(\frac{f_i}{\varsigma}\right) \sum_{j=1}^{H} \left[\frac{b_j \exp\left(-\frac{\mathcal{U}_{ij}}{\varsigma}\right)}{\sum_{k=1}^{N} \exp\left(\frac{f_k - \mathcal{U}_{kj}}{\varsigma}\right)} \right] = a_i \exp\left(-\frac{f_i + \zeta}{\tau}\right). \tag{25}$$

Specifically, $f_s^{(\ell+1)}$ can be updated at the ℓ-th iteration:

$$f_s^{(\ell+1)} = \kappa\varsigma \left[\log(a_s) - \frac{\zeta}{\tau} - \log\left(\sum_{j=1}^{H} \frac{b_j \exp\left(-\frac{\mathcal{U}_{sj}}{\varsigma}\right)}{\mathscr{M}_j} \right) \right] = \mathscr{U}_s\left(f_1^{(\ell)}, \cdots, f_s^{(\ell)}, \cdots, f_N^{(\ell)}\right) \tag{26}$$

where $\kappa = \tau/(\tau + \varsigma)$ and $\mathscr{M}_j = \sum_{k=1}^{N} \exp\left(\frac{f_k^{(\ell)} - \mathcal{U}_{kj}}{\varsigma}\right)$. The convergence of the WFPI algorithm can be guaranteed by considering the following results:

$$\frac{\partial \mathscr{U}_s}{\partial f_1^{(\ell)}} = -\frac{\kappa\varsigma}{\sum_{j=1}^{H} \frac{b_j \exp\left(-\frac{\mathcal{U}_{sj}}{\varsigma}\right)}{\mathscr{M}_j}} \frac{\partial}{\partial f_1^{(\ell)}} \left(\sum_{j=1}^{H} \left[\frac{b_j \exp\left(-\frac{\mathcal{U}_{sj}}{\varsigma}\right)}{\mathscr{M}_j} \right] \right)$$

$$= \frac{\kappa}{\sum_{j=1}^{H} \frac{b_j \exp\left(-\frac{\mathcal{U}_{sj}}{\varsigma}\right)}{\mathscr{M}_j}} \sum_{j=1}^{H} \left[\frac{b_j \exp\left(-\frac{\mathcal{U}_{sj}}{\varsigma}\right)}{\mathscr{M}_j} \cdot \frac{\exp\left(\frac{f_1^{(\ell)} - \mathcal{U}_{1j}}{\varsigma}\right)}{\mathscr{M}_j} \right] < 1 \tag{27}$$

Likewise we can obtain:

$$\sum_{i=1}^{N} \left| \frac{\partial \mathscr{U}_s}{\partial f_i^{(\ell)}} \right| = \frac{\kappa}{\sum_{j=1}^{H} \frac{b_j \exp\left(-\frac{\mathcal{U}_{sj}}{\varsigma}\right)}{\mathscr{M}_j}} \sum_{j=1}^{H} \left[\frac{b_j \exp\left(-\frac{\mathcal{U}_{sj}}{\varsigma}\right)}{\sum_{k=1}^{N} \exp\left(\frac{f_k^{(\ell)} - \mathcal{U}_{kj}}{\varsigma}\right)} \right] = \frac{\tau}{\tau + \varsigma} < 1 \tag{28}$$

That is, we can obtain the infinity-norm of Jacobian matrix is smaller than 1 and it indicates the convergence of WFPI. Finally, further optimize ζ by considering $\nabla_\zeta \mathcal{L}_{\mathrm{F}} = 0$. Specifically, we can obtain the exact value of ζ via:

$$\zeta = \tau \left[\log\left(\sum_{i=1}^{N} a_i \exp\left(-\frac{f_i}{\tau}\right) \right) - \log\left(\sum_{j=1}^{H} b_j \right) \right] \tag{29}$$

Apparently, we can easily verify that $\sum_{j=1}^{N} b_j = \sum_{i=1}^{N} a_i \exp(-(f_i + \zeta)/\tau)$ are satisfied during each iterations and indicates the importance of ζ in the optimization

process. The optimization algorithm of PUOT is provided in Alg.2. In summary, $\exp(-(f_i+\zeta)/\tau)$ can be regarded as the reweighted coefficient for the target data samples. Then we can further utilize weight barcode establishment to determine target unknown samples accordingly. The demo code[1] for PUOT is provided:

```
def puot(cost, src_weight, trg_weight,
         tau, eps=0.01, iter=50, tol=1e-6):
    num_src = cost.shape[0]
    f = np.ones((num_src,)) / num_src
    zeta = 0
    var = tau * eps / (tau + eps)
    for e in range(iter):
        prev_f = f
        temp = (f.reshape(-1,1)-cost)/eps
        den = np.sum(np.exp(temp), axis=0)
        term = np.exp(-cost/eps) / den
        sum_term = np.sum(term *
            trg_weight.reshape(1,-1),
            axis=1)
        v1 = np.log(src_weight)-zeta/tau
        f = var*(v1-np.log(sum_term))

        exp_f = np.exp(-f/tau)
        src_sum = np.sum(src_weight*exp_f)
        trg_sum = np.sum(trg_weight)
        zeta = tau*np.log(src_sum/trg_sum)
        if np.linalg.norm(prev_f-f)<=tol:
            break
    return src_weight*np.exp(-(f+zeta)/tau)
```

Listing 1. Python code for PUOT

C. Optimization on SSLM

In this section, we will provide the optimization details on solving SSLM with variables ψ and ϕ as shown below:

$$\max_{\psi,\phi} \mathcal{L}_{\mathrm{S}} = \sum_{i=1}^{N_{\mathrm{kno}}} \psi_i \widehat{a}_i + \sum_{j=1}^{H} \phi_j \widehat{b}_j - \frac{\epsilon}{2} \sum_{i=1}^{N_{\mathrm{kno}}} \sum_{j=1}^{H} \left[\frac{\psi_i + \phi_j - \widehat{C}_{ij}}{\epsilon} \right]_+^2, \tag{30}$$

We adopt Block Coordinate Descend (BCD) to optimize the above problem. Specifically, we first optimize ψ as:

$$\frac{\partial \mathcal{L}_{\mathrm{S}}}{\partial \psi_i} = 0 \quad \Rightarrow \quad \Psi(\psi_i) = \sum_{j=1}^{H} \left[\psi_i - (\widehat{C}_{ij} - \phi_j) \right]_+ = \epsilon \widehat{a}_i \tag{31}$$

After that we optimize ϕ as:

$$\frac{\partial \mathcal{L}_{\mathrm{S}}}{\partial \phi_j} = 0 \quad \Rightarrow \quad \Phi(\phi_j) = \sum_{i=1}^{N_{\mathrm{kno}}} \left[\phi_j - (\widehat{C}_{ij} - \psi_i) \right]_+ = \epsilon \widehat{b}_j \tag{32}$$

[1]Colab link for PUOT

Finally we can further calculate the optimal solution of $\gamma^{(l+1)}$ at the l-th iteration as shown:

$$\begin{cases} \gamma_{ij}^{(l+1)} = \max(0, [\psi_i^{(l)} + \phi_j^{(l)} + \epsilon\gamma_{ij}^{(l)} - C_{ij}]/\epsilon) \\ \widehat{C}_{ij}^{(l+1)} = C_{ij} - \epsilon\gamma_{ij}^{(l)} \end{cases} \tag{33}$$

After several iterations, we can obtain the optimal solution on ψ and ϕ, and then we can obtain the optimal solution on γ accordingly. The demo code[1] for SSLM is provided:

```
def SSLM(cost, src_weight, trg_weight,
         eps, n_iters=50):
    import ot
    Ns = cost.shape[0]
    Nt = cost.shape[1]
    pi = np.ones([Ns ,Nt]) / (Ns * Nt)
    for i in range(n_iters):
        pi = ot.smooth.smooth_ot_dual(
                src_weight,
                trg_weight,
                cost - eps * pi,
                reg = eps/2,
                reg_type='l2')
    return pi
```

Listing 2. Python code for SSLM

D. More Experiential Results

D.1. Setup

Office-31 [77] is the commonly-used computer vision dataset for domain adaptation with 4,652 images from three different domains: *Amazon* (**A**), *Webcam* (**W**) and *DSLR* (**D**). We adopt the same set of known and unknown classes following previous works [46]. We conduct the openset domain adaptation on the following six tasks: **A**→**W**, **A**→**D**, **D**→**W**, **W**→**D**, **D**→**A** and **W**→**A**.

Office-Home [84] is a standard benchmark dataset which includes 15,500 images in 65 object classes in office and home settings, forming four dissimilar domains: Artistic images (**Ar**), Clip Art (**Cl**), Product images (**Pr**) and Real-World (**Rw**). We select the first 25 classes as known and the rest 26-65 classes as unknown in alphabetic order following [46]. We conduct 12 openset tasks on *Office-Home* thorough evaluations: **Ar**↔**Cl**, **Ar**↔**Pr**, **Ar**↔**Rw**, **Cl**↔**Pr**, **Cl**↔**Rw**, and **Pr**↔**Rw**.

VisDA-2017 [70] consists of two domains with **Synthetic** and **Real** with 12 classes in common. **Synthetic** has 152,397 synthetic 2D renderings of 3D objects and **Real** has 55,388 real images. We construct openset domain adaptation task in **VisDA-2017** following [46].

Digits is the classical dataset for digit classification which contains three standard digit classification datasets: **MNIST** [40], **USPS**[31] and **SVHN** [67]. Each dataset

[1]Colab link for SSLM

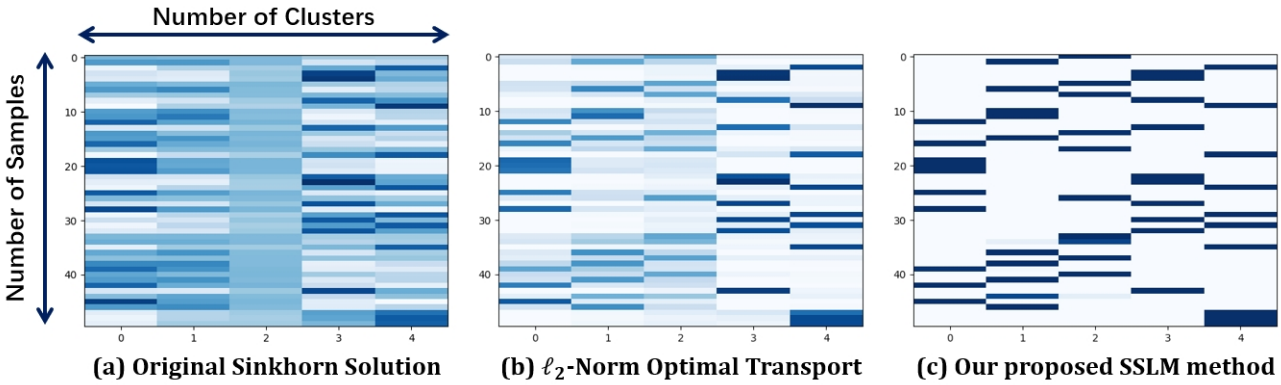

Figure 9. The illustration on Sparse Sample-Label Matching (SSLM) in sparse label assignment with toy examples. Sinkhorn and tradition ℓ_2-norm solvers could easily reach smooth but dense output solutions. Our proposed SSLM method can reach more sparse matching solution for better performance.

consists of 10 classes of digits, ranging from 0 to 9. We construct three open set domain adaptation tasks as **SVHN→MNIST**, **MNIST→USPS** and **USPS→MNIST** following the same evaluation protocol [78].