## A. IPOT

A detailed description of the IPOT algorithm used in our framework is summarized in Algorithm 2, where "$\odot$" is the Hadamard product and "$\div$" represents element-wise division. It is notable that this method works well with batch-based optimization, $i.e.$, $N$ in Algorithm 2 can represent either the size of the whole dataset or the size of a batch.

Specifically, compared with the Sinkhorn iteration algorithm, IPOT *changes* the objective function by adding an entropy regularizer on $\boldsymbol{T}$. Although such a modification converts the optimal transport problem to be strictly convex, its success is highly dependent on the choice of regularizer weight. On one hand, if the weight is too large, Sinkhorn iteration only obtains an over-smoothed $\boldsymbol{T}$ with a large number of iterations. On the other hand, if the weight is too small, Sinkhorn iteration suffers from numerical-stability issues. IPOT, in contrast, solves the original optimal transport problem. In particular, the regularizer in (7) just controls the learning process and its weight $\lambda$ mainly affects the convergence rate. If we reduce its weight $\lambda$ with respect to the number of iterations, the final result $\boldsymbol{T}^*$ will be equal to that obtained by solving (3) directly. Additionally, because the weight $\lambda$ mainly affects convergence rate, we can choose it in a wide range to achieve better numerical stability than Sinkhorn iteration.

**Algorithm 2:** IPOT Algorithm

```
1:  Input: Real features $\{\boldsymbol{x}_n\}_{n=1}^N$, generated features
    $\{\hat{\boldsymbol{x}}_m\}_{m=1}^M$, $\lambda = 0.5$, $\boldsymbol{\mu} = [\frac{1}{N}]$, $\boldsymbol{v} = [\frac{1}{M}]$.
2:  Output: Optimal transport $\boldsymbol{T}^*$
3:  Calculate $\boldsymbol{C} = [C_{nm}]$, with $C_{nm} = 1 - \frac{\boldsymbol{x}_n^\top \hat{\boldsymbol{x}}_m}{\|\boldsymbol{x}_n\|_2 \|\hat{\boldsymbol{x}}_m\|_2}$.
4:  $\boldsymbol{G} = \exp(-\frac{\boldsymbol{C}}{\lambda})$.
5:  Initialize $\boldsymbol{a} = \boldsymbol{\mu}$, $\boldsymbol{T}^{(1)} = \boldsymbol{\mu}\boldsymbol{v}^\top$
6:  for $t = 1, ..., T$ do
7:      $\boldsymbol{K} = \boldsymbol{G} \odot \boldsymbol{T}^{(t)}$
8:      Sinkhorn-Knopp Algorithm:
9:      for $j = 1, ..., J$ do
10:         $\boldsymbol{b} = \frac{\boldsymbol{v}}{\boldsymbol{K}^\top \boldsymbol{a}}$ and $\boldsymbol{a} = \frac{\boldsymbol{\mu}}{\boldsymbol{K}\boldsymbol{b}}$.
11:     end for
12:     $\boldsymbol{T}^{(t+1)} = \text{diag}(\boldsymbol{a})\boldsymbol{K}\text{diag}(\boldsymbol{b})$
13: end for
14: $\boldsymbol{T}^* = \boldsymbol{T}^{(T+1)}$
```