APPENDIX

A. Training Time Consumption

Table VI reports the training time consumption and AUC values of *SPSSOT* with different batch sizes. Though the optimal transport algorithm and the group entropic loss calculation have high complexity (super-quadratically with the size of the sample), the training usually takes only a few minutes because of the multiple rounds of minibatch iterative optimization [40] [41]. Therefore, we can find that as the batch size increases, the training time increases, but the AUC value does not change significantly. In other words, a larger batch size does not necessarily lead to a higher yield. Therefore, we choose 128 as the batch size of *SPSSOT*. At the same time, Table VII compares the training time of different semi-supervised transfer learning methods. The time consumption of our method is comparable to that of baselines. Considering that our method can achieve the best performance, such time consumption is generally acceptable in practice.

TABLE VI
TRAINING TIME CONSUMPTION WITH DIFFERENT BATCH SIZES.

Batch	MIMIC → Challenge		Challenge → MIMIC	
Size	AUC	Time(s)	AUC	Time(s)
64	63.73 ± 0.16	163.52	74.78 ± 0.35	148.74
128	**65.10 ± 0.24**	181.38	**76.05 ± 0.54**	167.31
256	64.45 ± 0.45	235.80	75.87 ± 0.32	220.82
512	64.46 ± 0.69	406.63	75.14 ± 0.73	392.36

TABLE VII
TRAINING TIME CONSUMPTION WITH DIFFERENT METHODS.

Method	MIMIC → Challenge		Challenge → MIMIC	
	AUC	Time(s)	AUC	Time(s)
MME	61.49 ± 0.84	75.28	75.07 ± 0.70	68.90
LIRR	62.76 ± 0.95	140.45	75.35 ± 0.59	138.64
S^3D	61.87 ± 0.61	165.82	75.56 ± 0.37	152.79
SPSSOT	**65.10 ± 0.24**	181.38	**76.05 ± 0.54**	167.31

B. Synchronous Self-paced Downsampling

In general, we want to downsample the samples without Sepsis to make the dataset more balanced. However, downsampling unlabeled data is non-trivial as we do not know their labels. In *SPSSOT*, we only consider obtaining balanced training data from the source and target labeled data. Here we further explore whether downsampling the unlabeled data is effective. We design a strategy to downsample the labeled and unlabeled data synchronously based on the widely-used stratified sampling technique [63]. The basic idea is to use the currently-trained model to predict unlabeled data, and then downsampling the unlabeled data according to prediction probabilities. In particular, we modify *SPSSOT* to achieve synchronous downsampling of labeled and unlabeled data in the self-paced ensemble process, named *S^2PSSOT*: (i) iterate 1000 times with all the data to obtain the initialized base classifier SSOT; (ii) obtain the prediction probability of 79% unlabeled data by the base classifier, split them into 10 bins according to prediction probabilities, and keep the proportion of downsampled unlabeled data in each bin is consistent with downsampled labeled data; (iii) iteratively train 1000 times with the downsampled data and go back to step (ii). We repeat steps (ii) & (iii) five times for getting the final model. The detailed algorithm flow is shown in Algorithm 3 (line 13 to 17 is to downsample the target unlabeled data).

Algorithm 3 Semi-supervised Optimal Transport with Synchronous Self-paced Ensemble (*S^2PSSOT*)

Require: Source data as $\mathcal{D}^s = \{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^{n_s}$; Target labeled data as $\mathcal{D}^l = \{(\boldsymbol{x}_j^l, y_j^l)\}_{j=1}^{n_l}$; Target unlabeled data as $\mathcal{D}^u = \{(\boldsymbol{x}_k^u)\}_{k=1}^{n_u}$;Hardness function $\mathcal{H}$; Base classifier *SSOT*; Number of base classifiers n; Number of hardness bins k; Number of probability bins m; Total number of training iterations of *SSOT* T;

1: Initialize $SSOT_0$ according to Algorithm 1;
2: **for** $i = 1$ to n **do**
3: Ensemble $F_i(\mathcal{D}^s, \mathcal{D}^l, \mathcal{D}^u) = \frac{1}{i}\sum_{j=0}^{i-1} SSOT_j(\mathcal{D}^s, \mathcal{D}^l, \mathcal{D}^u)$;
4: **for** $\mathcal{D} \in \{\mathcal{D}^s, \mathcal{D}^l\}$ **do**
5: Initialize $\mathcal{P} \Leftarrow$ minority in $\mathcal{D}$;
6: Cut majority set into k bins w.r.t. $\mathcal{H}(\mathcal{D}, F_i)$: $B_1, B_2, \cdots, B_k$;
7: Average hardness contribution in l-th bin: $h_l = \sum_{m \in B_l} \mathcal{H}(x_m, y_m, F_i)/|B_l|, \forall l = 1, \cdots, k$;
8: Update self-paced factor $\omega = tan(\frac{i\pi}{2n})$;
9: Unnormalized sampling weight of l-th bin: $p_l = \frac{1}{h_l + \omega}$, $\forall l = 1, \cdots, k$;
10: Downsample from l-th bin with $\frac{p_l}{\sum_m p_m} \cdot |\mathcal{P}|$;
11: **end for**
12: Obtain the downsampled labeled subset $\{\mathcal{D}_d^s, \mathcal{D}_d^l\}$;
13: Calculate the probabilities: $P_d^l = F_i(\mathcal{D}_d^l)$ and $P^u = F_i(\mathcal{D}^u)$;
14: Cut $\mathcal{D}_d^l$ into m bins according to $P_d^l : G_1^l, G_2^l, \cdots, G_m^l$;
15: Cut $\mathcal{D}^u$ into m bins according to $P^u : G_1^u, G_2^u, \cdots, G_m^u$;
16: Calculate the percentage of each bin in $\mathcal{D}_d^l$: $g_j = |G_j^l|/|\mathcal{D}_d^l|$;
17: Downsample from j-th bin, G_j^u, with $g_j \cdot |\mathcal{D}^u|$;
18: Train $SSOT_i$ using $\{\mathcal{D}_d^s, \mathcal{D}_d^t, \mathcal{D}_d^u\}$ according to Algorithm 1;
19: **end for**
20: **return** Final ensemble model $F(\mathcal{D}^s, \mathcal{D}^l, \mathcal{D}^u) = \frac{1}{n}\sum_{m=1}^{n} SSOT_m(\mathcal{D}^s, \mathcal{D}^l, \mathcal{D}^u)$;

As shown in Table VIII, there is no significant improvement of the new *S^2PSSOT* compared to the original *SPSSOT*. The possible reason is that the prediction probabilities of the unlabeled data still have uncertainties and thus the prediction-probability-based unlabeled data downsampling may not achieve the ideal data balancing effect. We believe this is an open and interesting question worthy of further exploration.

TABLE VIII
RESULTS OF SYNCHRONOUS DOWNSAMPLING FROM TARGET UNLABELED DATA.

Method	MIMIC → Challenge	Challenge → MIMIC
SPSSOT	65.10 ± 0.24	76.05 ± 0.54
S^2PSSOT	64.89 ± 0.28	75.34 ± 0.39

C. Analysis of Outlier Disturbance

The self-paced sampling in *SPSSOT* has filtered out some noise samples through self-paced hardness harmonization. In general, the outliers would not affect the calculation of class centers. To confirm this, we also use a popular outlier detection algorithm, the isolation forest algorithm [64], to filter out the

outliers before calculating the class centers. As shown in Table IX, adding an explicit step of outlier removal has no noticeable effect on the results. Thus, as expected, the outliers do not seriously affect the accuracy of the calculation of class centers in *SPSSOT*.

TABLE IX
RESULTS OF REMOVING OUTLIERS.

Method	MIMIC → Challenge	Challenge → MIMIC
SPSSOT	65.10 ± 0.24	76.05 ± 0.54
+ outlier removal	65.00 ± 0.20	75.89 ± 0.35

D. Selection of ρ in Label Adaptive Constraint

In Eq. (3), we adapt a parameter, ρ, to adjust the transport cost between two samples with the same label; especially when $\rho = 0$, the transport cost is 0; when $\rho = 1$, the transport cost is calculated only according to the similarity of features (same as the unsupervised setting). We set $\rho = \{0, 0.05, 0.1, 0.2, 0.4\}$ and conduct experiments. The results are shown in Table X. It can be observed that when ρ is small (between 0 to 0.1), the performance is better and relatively stable; then as ρ increases, the AUC shows a slow downward trend. This indicates that in our task, it is better to set a small value to ρ, and setting $\rho = 0$ (i.e., ignoring the transport cost if two samples have the same label) is also reasonable. In *SPSSOT*, we set ρ to 0.1 and 0.05 for MIMIC → Challenge and Challenge → MIMIC, respectively.

TABLE X
RESULTS OF DIFFERENT $\boldsymbol{\rho}$.

ρ	MIMIC → Challenge	Challenge → MIMIC
0	64.98 ± 0.26	75.96 ± 0.68
0.05	64.99 ± 0.35	**76.05 ± 0.54**
0.1	**65.10 ± 0.24**	75.90 ± 0.52
0.2	64.47 ± 0.39	74.75 ± 1.15
0.4	63.91 ± 0.21	74.19 ± 0.75

E. Unmatched Features

In *SPSSOT*, we use only the features shared by two domains (listed in Table I) with a domain-shared feature generator $\mathcal{G}$. Here, we list the (unmatched) private features of two datsets in Table XI. Considering that our task is a transfer learning setting, we discuss the private features for the target domain and source domain separately.

1) Target private features: Considering target private features may be helpful to the target classification task, we design new network structures to incorporate these features (as shown in Fig. 10): (i) add a feature encoder $\mathcal{G}_{pri}$ for private features (the structure is the same as $\mathcal{G}$); (ii) concatenate the output of $\mathcal{G}_{pri}$ and the output of $\mathcal{F}$'s first layer; (iii) take the concatenation as the input of a new target classifier $\mathcal{F}_{new}$. After training *SPSSOT*, we transfer the parameters of *SPSSOT* and randomly initialize parameters in other components, and then update parameters with the target labeled data. In brief, we finetune *SPSSOT* by the target labeled data with full features (i.e., shared and private features).

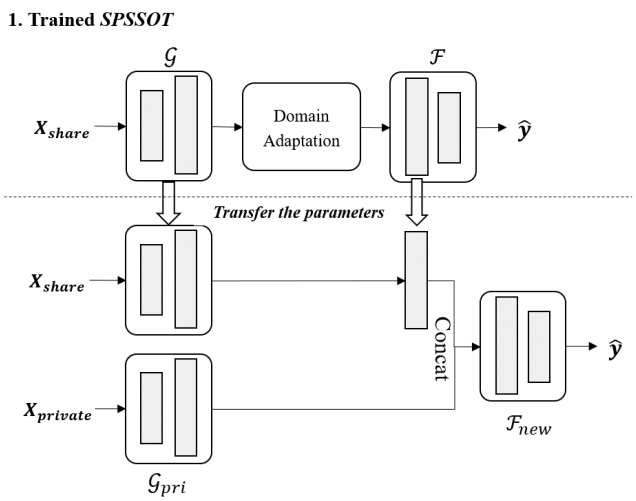

Fig. 10. The network structure to transfer *SPSSOT*'s parameters to target domain with private features. $\boldsymbol{X_{share}}$ means only using shared features as the input, similarly, $\boldsymbol{X_{private}}$ means only using target private features as input.

As illustrated in Table XII, we can find that there is a significant improvement in Challenge → MIMIC but no significant change in MIMIC→Challenge. This may be because Challenge only has two private features which are not important.

TABLE XI
THE PRIVATE FEATURES OF TWO DATASETS.

MIMIC	Challenge
Height, Weight, GCS, CRP, PCT, D-Dimer, FBG, TCO_2	TBil(Total bilirubin), Troponin I

TABLE XII
RESULTS OF ADDING TARGET PRIVATE FEATURES.

Method	MIMIC → Challenge	Challenge → MIMIC
SPSSOT	65.10 ± 0.24	76.05 ± 0.54
+ $fea^T_{private}$	64.88 ± 0.51	77.53 ± 0.59

2) Source private features: Transferring the knowledge from source private features for the prediction in the target domain is non-trivial. The optimal transport technique is hard to directly apply to source private features, as no corresponding features exist in the target domain (so feature similarity cannot be appropriately calculated between a source sample and a target sample). To address this issue, it may require incorporating more transfer learning techniques, e.g., knowledge distillation [61].