Cross-hospital Sepsis Early Detection via Semi-supervised Optimal Transport with Self-paced Ensemble

Ruiqing Ding, Yu Zhou, Jie Xu, Yan Xie, Qiqiang Liang, He Ren,
Yixuan Wang, Yanlin Chen, Leye Wang and Man Huang

***Abstract*— Leveraging machine learning techniques for Sepsis early detection and diagnosis has attracted increasing interest in recent years. However, most existing methods require a large amount of labeled training data, which may not be available for a target hospital that deploys a new Sepsis detection system. More seriously, as treated patients are diversified between hospitals, directly applying a model trained on other hospitals may not achieve good performance for the target hospital. To address this issue, we propose a novel semi-supervised transfer learning framework based on optimal transport theory and self-paced ensemble for Sepsis early detection, called *SPSSOT*, which can efficiently transfer knowledge from the source hospital (with rich labeled data) to the target hospital (with scarce labeled data). Specifically, *SPSSOT* incorporates a new optimal transport-based semi-supervised domain adaptation component that can effectively exploit all the unlabeled data in the target hospital. Moreover, self-paced ensemble is adapted in *SPSSOT* to alleviate the class imbalance issue during transfer learning. In a nutshell, *SPSSOT* is an end-to-end transfer learning method that automatically selects suitable samples from two domains (hospitals) respectively and aligns their feature spaces. Extensive experiments on two open clinical datasets, MIMIC-III and Challenge, demonstrate that *SPSSOT* outperforms state-of-the-art transfer learning methods by improving 1-3% of AUC.**

***Index Terms*— Semi-supervised Transfer Learning, Sepsis Early Detection, Optimal Transport Theory**

This work was supported by grants from the National Natural Science Foundation of China (No. 81801940 Y.Z.).

Ruiqing Ding and Leye Wang are with Key Lab of High Confidence Software Technologies (Peking University), Ministry of Education, China, and also with Computer Science School, Peking University, Beijing 100871, China (e-mail: ruiqing@stu.pku.edu.cn, leyewang@pku.edu.cn).

Yu zhou, Qiqiang Liang and Man Huang are with General Intensive Care Unit, Zhejiang University School of Medicine Second Affiliated Hospital, Hangzhou 310009, Zhejiang, China (e-mail:{naseph, deter_leung, huangman}@zju.edu.can).

Jie Xu is with IT center, Zhejiang University School of Medicine Second Affiliated Hospital, Hangzhou 310009, Zhejiang, China (e-mail:2202113@zju.edu.can).

Yan Xie and He Ren are with Beijing HealSci Technology Co., Ltd., Beijing 100176, China (e-mail: {yan.xie, he.ren}@healscitech.com).

Yixuan Wang is with Department of Computer Science and Technology, Peking University, Beijing 100871, China (e-mail: kugamashiro@pku.edu.cn).

Yanlin Chen is with School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China (e-mail: 1391469597@qq.com).

(Equal Contribution: Ruiqing Ding and Yu Zhou, Corresponding author: Man Huang.)

I. Introduction

SEPSIS is a life-threatening disease that occurs when the body's response to infection is out of balance [1]. In severe cases, it will trigger body changes that may damage multiple organ systems and lead to death [2]. Sepsis has become a major cause of in-hospital death for intensive care unit (ICU) patients, which places an enormous burden on public health expenditures [3] [4]. In 2013, Sepsis was responsible for 10% of the ICU admissions and occupied about 25% of the ICU beds in US hospitals, accounting for over $23.6 billion (6.2%) of total US hospital costs [5]. Early detection is crucial to the sepsis management; with each one-hour delay in the administration of antibiotic treatment, the mortality rate increases by 7% [6].

Recently, machine learning techniques start to be applied in Sepsis diagnosis and early detection, such as the linear model [7], Support Vector Machine [8], Neural Network [9], Gradient Boosting Decision Tree [10]. These methods need huge amounts of labeled training data to ensure performance. In reality, one hospital may hold its treated patients' Electronic Medical Records (EMRs), but it is common that most EMRs are not properly labeled for a machine learning task (e.g., Sepsis early detection) [11]. Therefore, how to use these (unlabeled) records to predict the health situations of new patients is an important problem to be addressed.

Transfer learning [12] is a promising machine learning paradigm for the label-scarcity scenario; it provides an unconventional perspective to transfer external knowledge from another hospital with rich labeled data to improve the machine learning performance of a target hospital with scarce labeled data. It also reduces expensive data-labeling costs. The state-of-the-art transfer learning strategy is fine-tuning [13] [14]; however, overfitting is often caused by fine-tuning a large number of parameters with very small labeled data [15]. Consequently, many challenges still exist for successful knowledge transfer, especially in clinical data:

Covariate Shift. Different medical devices in different hospitals may result in diverse test values. Also, patients' agglomeration factor cannot be neglected. Specifically, patients tend to choose a hospital that is more appropriate for their diseases and health conditions [16]. Hence, the sets of patients' information collected from two hospitals are often different

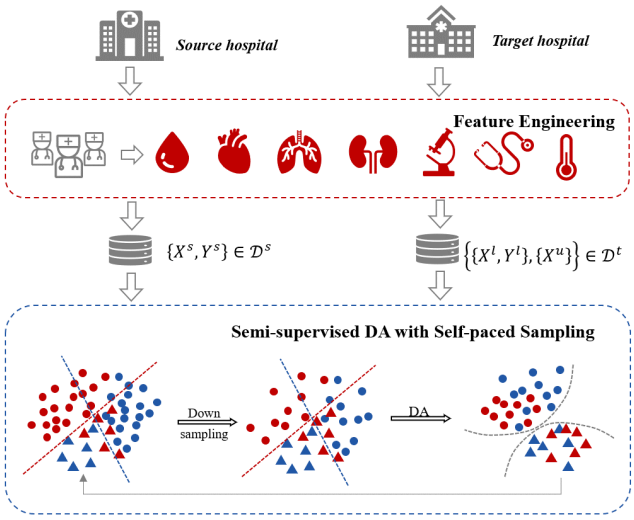


Fig. 1. The Overall Framework of Semi-supervised Optimal Transport with Self-paced Ensemble (*SPSSOT*), which consists of 3 main parts: (1) Feature Engineering to extract Sepsis-related features under the guidance of doctors; (2) Self-paced sampling to filter out "more contributing" samples from negative samples (no Sepsis, shown as circles), which will help to improve the performance of classifier. (3) DA (Semi-supervised domain adaptation): given all data with labels of source hospital (shown as red), little data with labels and most data without labels in target hospital (shown as blue), align two feature spaces via optimal transport theory and learn a better classifier to distinguish whether Sepsis will occur. *Best viewed in color.*

from each other; in other words, they are not in the same feature space. Consequently, it is essential to map them into a common hidden space, known as *domain adaptation* [17].

Label shift. Label shift implies that the label distribution changes from the source to the target [18]. In particular, the incidence of a disease may fluctuate with locations and time, easily leading to a negative transfer. To alleviate this pitfall, prior methods propose to re-weight source samples' importance [19]; however, they incur a huge computational burden when a large number of samples exist.

Class Imbalance. Imbalanced data is ubiquitous especially for medical diagnostic datasets [20], and it exhibits a long-tailed distribution [21]. During transfer learning, it is also vital to reduce the classification bias caused by data imbalance and find more appropriate decision boundaries.

To overcome the above difficulties, we propose a semi-supervised transfer learning approach based on optimal transport [22] and self-paced ensemble [23] approach to complete cross-hospital Sepsis early detection. There are **three main components** of *SPSSOT*: *feature engineering* under the guidance of doctors to extract features associated with Sepsis, *self-paced ensemble* to achieve data balance, and *semi-supervised domain adaptation via optimal transport* to tackle the problem of inconsistent feature spaces. The overall framework is shown in Fig. 1. Our contributions are summarized as follows:

(1) To the best of our knowledge, this is the first work on cross-hospital Sepsis early detection. In particular, by properly transferring knowledge from another hospital with rich labeled data, our method can enable good detection performance for the target hospital with little labeled data.

(2) Considering the inconsistent feature distributions and the unbalanced noisy data status in cross-hospital Sepsis early detection, we propose a novel end-to-end deep transfer learning framework, called *SPSSOT*, consisting of three components: feature engineering, semi-supervised domain adaptation with optimal transport, and self-paced ensemble. More specifically, in *semi-supervised domain adaptation with optimal transport*, we design a label-adaptive optimal transport strategy to achieve the precise-pair-wise optimal transport, and an intra-domain deep feature discrimination strategy to find a better decision boundary. In *self-paced ensemble*, we improve and incorporate an ensemble algorithm for imbalanced classification [23] into our deep transfer learning framework, which can adaptively downsample the majority data from both domains to alleviate the class imbalance issue.

(3) By conducting the experiments on mutual transfer between two open clinical datasets, MIMIC and Challenge, we have validated *SPSSOT* to improve the AUC values by at least 3% and 1% with only 1% labeled data in the target domain compared to state-of-the-art transfer learning methods [24]–[26].

II. Related Work

This paper mainly provide a new solution for Sepsis early detection when there are few labeled EMRs in the hospital. We propose a transfer learning framework based on optimal transport theory [27] to cope with the data discrepancy between different hospitals; and introduce a self-paced ensemble method to overcome the extreme label imbalance problem. Accordingly, we provide a brief overview of related work in four fields, i.e., Sepsis early detection, transfer learning with optimal transport, data imbalance, and self-paced learning.

Sepsis Early Detection. Machine learning (ML) techniques excel in the analysis of complex signals in data-rich environments which promise to improve the early detection of Sepsis. Most of the studies are carried out in the ICU [11] [28], and some of them are specifically on neonatal Sepsis [29] [30]. Systematic review and meta-analysis indicate that individual machine learning models can accurately predict the onset of Sepsis in advance on retrospective data [10] [31]. The PhysioNet/Computing in Cardiology (CinC) Challenge 2019 focused on this issue and promoted the development of open-source AI algorithms for real-time and early detection of Sepsis [32]. Such approaches, which typically apply ML techniques to clinical data, can dynamically suggest real-time predictions and optimal treatments for Sepsis patients and yield excellent results in the medical field. However, the variety of studies engaged in Sepsis early detection without sufficient labeled data remains small.

Transfer Learning with Optimal Transport. The core of transfer learning is to align the source and target distributions by minimizing a divergence that measures the discrepancy between them. Optimal Transport (OT) theory can be regarded as one of the discrepancy-based alignment methods, as it can be used for calculating Wasserstein distances between probability distributions [33]. Given the cost function (e.g., l_2 distance)

between samples in the source and target domains, we can calculate the probabilistic coupling matrix γ. It has been applied in domain adaptation to learn the transformation between different domains [34] [35], with associated theoretical guarantees [36]. Moreover, it is applicable to different transfer scenarios, including unsupervised [37] and semi-supervised [38] situations. Initially, limited by the space complexity of OT (super-quadratically with the size of the sample), it can only be deployed to tackle problems of small or medium size [39]. Recently, more and more work has attempted to combine deep learning method with OT to train through multiple rounds of minibatch iterative optimization, such as DeepJDOT [40] and RWOT [41], breaking the limit of complexity. In our setting, there are few labeled patients in the target hospital, which can be viewed as a problem of semi-supervised transfer learning. In contrast to the common approach in the unsupervised case, we can further consider the coupling constraints for labeled samples when using OT.

Data Imbalance. Traditionally, ML algorithms may assume that the number of samples in considered classes are roughly the same, which is not the case in real-life problems. In many medical datasets, the ratio of minority class to majority class can be 1:10, even up to 1:50 [20]. The key point of imbalanced learning is that the minority classes are often more important, namely, we need to focus on the diseased samples rather than the healthy samples. A series of studies have been conducted to overcome data imbalance issue, which can be classified into three types: i) data-level methods, which adjust the dataset to balance the minority and majority. A typical way is to downsample the majority or oversample the minority; ii) algorithm-level methods, which do not change the dataset directly, but rather enhance the attention of the model to the minority by modifying the algorithm, i.e., by setting a cost matrix in cost-sensitive learning with help from domain experts [42]; iii) the combination of both, which takes the advantage of the above methods. For instance, SMOTEBoost [43] combines SMOTE [44] with Boosting [45] ensemble learning to gain a strong ensemble classifier, SPE [23] tries to handle tasks on the highly imbalanced, noisy and large-scale dataset by introducing the "classification hardness" function and undersampling with an iterated strategy.

Self-paced Learning. It is a learning paradigm to generates the sequence of training samples by the learner itself, whose core idea is to adaptively select the most informative samples in each iteration [46]. In recent years, the self-paced learning regime has been adopted for various tasks, including weakly supervised object detection [47], co-saliency detection [48], and data imbalance [23], which indicates the effectiveness of such a learning paradigm. Though there are some studies that have combined self-paced learning with deep learning for joint learning [49], we try to combine self-paced learning with optimal transport to eliminate the impact of data imbalance on semi-supervised domain adaptation.

III. Preliminaries

In this section, we first define our research problem from an application perspective. Then, we abstract the problem in a transfer learning setting.

A. Sepsis Early Detection

The objective is to use patients' demographic and physiological data for Sepsis early detection. Considering the early warning of Sepsis is potentially life-saving, we will detect sepsis 6 hours before the clinical diagnosis of Sepsis. This setting is consistent with the PhysioNet Computing in Cardiology Challenge 2019 [32] [50], whose topic is *Early Prediction of Sepsis from Clinical Data*.[1]

In short, given a set of n patients' clinical variables since they entered the ICUs, $\{\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_n\}$, where the i-th patient's is $\mathcal{X}_i = \langle \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m \rangle$, $\boldsymbol{x}_j$ is the clinical features of j-th time windows (we set the length of one time window as 6 hours). Then we aim to predict whether Sepsis will occur in the next 6 hours for each $\boldsymbol{x}_j$. Thus, it can be seen as a binary classification problem. The clinical variables will be explained in detail in Sec. IV-A.

B. Semi-supervised Transfer Learning Formulation

When we try to build the detection model in a target hospital with few labeled data, the basic idea is to learn knowledge from other rich data sources. In other words, we can consider this problem as semi-supervised transfer learning.

In particular, we are given a source domain and a target domain with the same features. The source domain contains a large number of labeled samples, and the target domain only contains a limited number of labeled samples (i.e. most samples are unlabeled). The task is to improve the classification accuracy in the target domain. We denote the source domain as $\mathcal{D}^s = \{(\boldsymbol{x}_i^s, y_i^s)|i = 1, 2, \cdots, n_s\}$, $\boldsymbol{x}_i^s \in \mathbb{R}^{d_s}$, the target domain as $\mathcal{D}^t = \{\mathcal{D}^l, \mathcal{D}^u\}$ where the labeled data $\mathcal{D}^l = \{(\boldsymbol{x}_j^l, y_j^l)|j = 1, 2, \cdots, n_l\}$ and the unlabeled data $\mathcal{D}^u = \{(\boldsymbol{x}_k^u)|k = 1, 2, \cdots, n_u\}$, $\boldsymbol{x}_j^l, \boldsymbol{x}_k^u \in \mathbb{R}^{d_t}$. n_l and n_u are the number of labeled and unlabeled target samples, respectively, $n_t = n_l + n_u$ ($n_l \ll n_u$). We suppose $d_s = d_t$ (the source and target domains share the same features) and $\mathcal{Y}^s = \mathcal{Y}^t = \{0, 1\}$ (binary classification task, 1/0 indicates that Sepsis would/wouldn't happen in 6 hours).

IV. Methodology

In this section, we propose a semi-supervised transfer learning framework, *SPSSOT*, to address our research problem, whose schematic diagram is illustrated as Fig. 1. It consists of three main parts: (1) Feature Engineering, (2) Semi-supervised Optimal Transport, and (3) Self-paced ensemble.

A. Feature Engineering

We extract the clinical variables and Sepsis criteria from the Electronic Medical Records (EMRs). For each patient, 34 clinical variables are constructed, including 7 vital sign variables, 23 laboratory variables , and 4 demographic variables. Detailed variables are listed in Table I. The Sepsis-3 criteria are extracted as suspected infection with associated organ dysfunction (SOFA $\geq$ 2) [1] [51].

In particular, prior work has shown that typical vital signs, such as heart rate (HR), oxygen saturation (O_2Sat), body

[1] https://physionet.org/content/challenge-2019/1.0.0/

TABLE I
FEATURE DESCRIPTION

Measurement	Description
Vital sign variables	
HR	Heart rates (beats per minute)
O_2Sat	Pulse oximetry (%)
Temp	Temperature ($^\circ C$)
SBP	Systolic BP (mm Hg)
MAP	Mean arterial pressure (mm Hg)
DBP	Diastolic BP (mm Hg)
Resp	Respiration rate (breaths per minute)
Laboratory variables	
BaseExcess	Excess bicarbonate (mmol/L)
HCO_3	Bicarbonate (mmol/L)
FiO_2	Fraction of inspired oxygen (%)
pH	pH value
$PaCO_2$	The partial pressure of carbon dioxide from arterial blood (mm Hg)
SaO_2	Oxygen saturation from arterial blood (%)
AST	Aspartate transaminase (IU/L)
BUN	Blood urea nitrogen (mg/dL)
Alkalinephos	Alkaline phosphatase (IU/L)
Calcium	Calcium (mg/dL)
Chloride	Chloride (mmol/L)
Creatinine	Creatinine (mg/dL)
Bilirubin direct	Direct bilirubin (mg/dL)
Glucose	Serum glucose (mg/dL)
Lactate	Lactic acid (mg/dL)
Magnesium	Magnesium (mmol/dL)
Phosphate	Phosphate (mg/dL)
Potassium	Potassiam (mmol/L)
Hct	Hematocrit (%)
Hgb	Hemoglobin (g/dL)
PTT	Partial thromboplastin time (seconds)
WBC	Leukocyte count (count/L)
Platelets	Platelet count (count/mL)
Demographic variables	
Age	Age (years)
Sex	Female (0) or male (1)
HospAdmTime	Hours from hospitalization to ICU admission
ICULOS	Length of stay in ICU (hours since admission to ICU)

temperature (Temp), mean arterial blood pressure (MAP), and respiratory rate (Resp), would impact the Sepsis early detection over time [32] [52]. Besides, Sepsis incidence rate also varies with respect to the ICULOS (i.e., time stay in ICU). At the early phase, the incidence rate is moderate and slightly increases probably due to the patient prior conditions; at the middle phase, the incidence rate drops a little and becomes stable; at the late phase, the incidence rate increases drastically because there is big vulnerability for the patients that stay long in the ICU [10].

To capture the time series fluctuation, we take 6 hours as a time window. In the time slot, we calculate the maximum values, minimum values, means, standard deviations and number of non-missing for all vital signs and laboratory values, while keeping the latest values. Finally we concatenate these statistics with demographic variables as the final features to predict whether Sepsis will occur in the next 6 hours.

B. Semi-supervised Optimal Transport

Optimal transport theory is a promising strategy applied in transfer learning research. Most existing studies leverage it in unsupervised transfer learning [33]. That is, they assume that no labeled data is in the target domain. However, few labeled samples in the target domain are more in line with the real situation [25]. For instance, it is usually acceptable to label a few samples when we want to deploy a Sepsis early detection system in a new hospital, if this can significantly improve the system performance.

With this in mind, we design a novel semi-supervised optimal transport (*SSOT*) strategy for transfer learning. The purpose is to increase classification performance by minimizing the distribution discrepancy between the source and target domains with a well-structured neural network. Specifically, the neural network enables the end-to-end training of a transferable feature generator and an adaptive classifier, as illustrated in Fig. 2. Because clinical features can be viewed as tabular data, we choose classical multi-layer perception (MLP) with shared weights as the feature generator $\mathcal{G}$.

The main parts of *SSOT* are *label adaptive optimal transport*, *group entropic loss for unlabeled samples* and *intra-domain discriminative feature clustering*. In the following sections, we will present the details of *SSOT*.

1) Label Adaptive Optimal Transport: In the traditional optimal transport strategy for unsupervised transfer learning, the source and target samples are mapped to a shared feature space where the samples of both domains cannot be differentiated [33]. In semi-supervised transfer learning, there are *a few labeled samples* in the target domain; we then need to propose a new optimal transport strategy to effectively leverage these labeled target data. Therefore, beyond the traditional optimal transport strategy, our mechanism further conducts the optimization to ensure that *the labeled target samples should only be matched with the same-labeled source samples.*

Optimal Transport. The optimization of optimal transport is based on Kantorovich problem [53] which seeks for a general coupling $\gamma \in \mathcal{X}(\mathcal{D}^s, \mathcal{D}^t)$ between $\mathcal{D}^s$ and $\mathcal{D}^t$:

$$\gamma^* = \underset{\gamma \in \mathcal{X}(\mathcal{D}^s, \mathcal{D}^t)}{\arg\min} \int_{\mathcal{D}^s \times \mathcal{D}^t} \mathcal{C}(x^s, x^t) d\gamma(x^s, x^t) \tag{1}$$

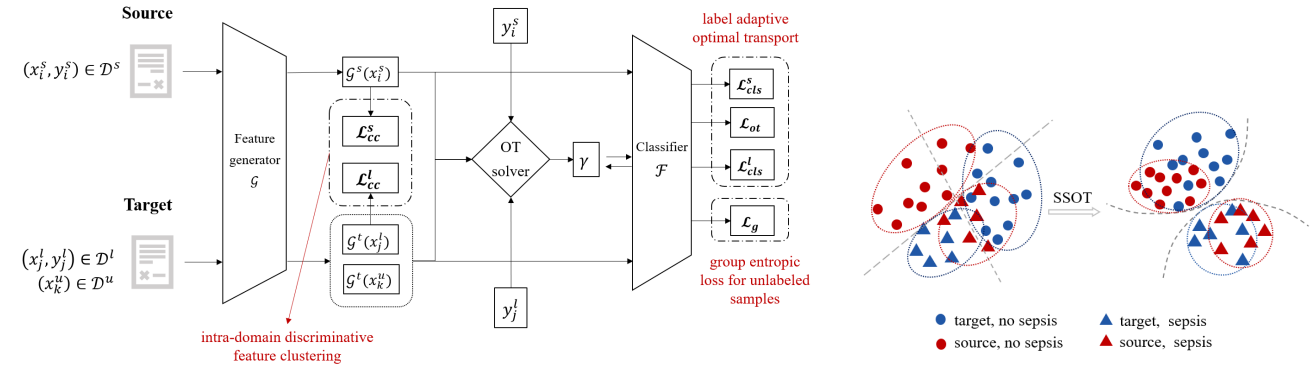
where $\mathcal{X}(\mathcal{D}^s, \mathcal{D}^t)$ denotes the probability distribution between $\mathcal{D}^s$ and $\mathcal{D}^t$.

The discrete reformulation is

$$\gamma^* = \underset{\gamma \in \mathcal{X}(\mathcal{D}^s, \mathcal{D}^t)}{\arg\min} \langle \gamma, \mathcal{C} \rangle_F \tag{2}$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius dot product, $\mathcal{C} \in \mathbb{R}^{n_s \times n_t}$ is the cost function matrix. $\mathcal{C}(x^s, x^t) = \|x^s - x^t\|^k$ represents the cost to move probability mass from x^s to x^t; we set $k = 2$ following the literature [33].

Label Adaptive Constraint. As a few data can be labeled in the target domain, we adjust the cost of transport according to the labels of the two domains' samples. If two samples have the same labels, it means that the transport cost is very low between these two samples. Therefore, we can use a parameter, ρ, to adjust the cost, i.e, $\mathcal{C}(x^s, x^t) = \rho$, if $y(x^s) = y(x^t)$; otherwise, we set the cost to 1. At the same time, for unlabeled target samples, we can consider supplementing a weight for transport cost to measure the difference between the predicted probabilities and the labels of source samples. Accordingly,



(a) The Architectures of *SSOT*: (1) Initialize the feature generator $\mathcal{G}$ and the classifier $\mathcal{F}$; (2) Fix $\mathcal{G}$ and $\mathcal{F}$, find the current best coupling $\hat{\gamma}$ between $\mathcal{D}^s$ and $\mathcal{D}^t(\{\mathcal{D}^l, \mathcal{D}^u\})$ by the OT solver, then fix $\hat{\gamma}$ and update the parameters of $\mathcal{G}$ and $\mathcal{F}$; (3) Iterative training.

(b) The Objectives of SSOT: (1) the marginal distributions of two domains are identical; (2) the samples belonging to the same class are more aggregated.

Fig. 2. Semi-supervised Domain Adaptation with Optimal Transport (*SSOT*).

Algorithm 1 Semi-supervised Optimal Transport (*SSOT*)

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 $\mathcal{D}^u = \{(\boldsymbol{x}_k^u)\}_{k=1}^{n_u}$; T is set as the total number of training iterations; n represents the batch-size for training.

Initialize the feature generator $\mathcal{G}$ and the classifier $\mathcal{F}$ by fine tuning;
for $i = 1$ to T **do**
  Randomly select half of source samples and target labeled samples;
  Calculate the class centers in two domains according to Eq.9.
  Randomly choose source samples $\{(x_i^s, y_i^s)\}_{i=1}^{n} \in \mathcal{D}^s$, target labeled samples $\{(x_j^l, y_j^l)\}_{j=1}^{n/2} \in \mathcal{D}^l$, and target unlabeled samples $\{(x_k^u)\}_{k=1}^{n/2} \in \mathcal{D}^u$;
  Fix $\hat{\mathcal{G}}$ and $\hat{\mathcal{F}}$, solve for γ;
  Fix $\hat{\gamma}$, update parameters of $\mathcal{G}$ and $\mathcal{F}$;
end for
return $\mathcal{G}$ and $\mathcal{F}$;

we design a reweight matrix, called *label adaptive matrix* $\mathcal{R}$. Then, the label adaptive optimal transport can be written as

$$\gamma^* = \underset{\gamma \in \mathcal{X}(\mathcal{D}^s, \mathcal{D}^t)}{\arg\min} \langle \gamma, \mathcal{R} \cdot \mathcal{C} \rangle_F \tag{3}$$

where

$$\mathcal{R}(x^t, x^s) = \begin{cases} \rho + (1-\rho) \cdot |y(x^s) - y(x^t)| \in \{\rho, 1\}, (x^t, y^t) \in \mathcal{D}^l \\ \rho + (1-\rho) \cdot |y(x^s) - \hat{y}(x^t)| \in [\rho, 1], x^t \in \mathcal{D}^u \end{cases},$$

$y(x)$ is the label of a sample x and $\hat{y}(x)$ is the prediction probability $P(y(x) = 1)$.

In summary, the solution to this problem can be described to minimize the following objective function

$$\mathcal{L}_{lot} = \alpha \mathcal{L}_{ot} + \theta_s \mathcal{L}_{cls}^s + \mathcal{L}_{cls}^l \tag{4}$$

where α and θ_s are hyper-parameters, $\mathcal{L}_{ot}$ is the cost of optimal transport, $\mathcal{L}_{cls}^s$ and $\mathcal{L}_{cls}^l$ are the cross entropy function of the source and target domain, i.e.,

$$\mathcal{L}_{ot} = \sum_{i,j} \gamma_{i,j}^* (\|\mathcal{G}(x_i^s) - \mathcal{G}(x_j^t)\|^2) \tag{5}$$

$$\begin{aligned} \mathcal{L}_{cls}^s &= -\sum_{x_i^s \in X_s} y(x_i^s) \log \hat{y}(x_i^s) \\ \mathcal{L}_{cls}^l &= -\sum_{x_j^l \in X_l} y(x_j^l) \log \hat{y}(x_j^l) \end{aligned} \tag{6}$$

2) Group Entropic Loss for Unlabeled Samples: It is insufficient to ensure that the mappings of source and target samples cannot be differentiated in a shared feature space. This only implies that the marginal distributions of the two domains are identical. To further mitigate the differences in the conditional distributions, we borrow the labels of the source domain to calculate the classification loss of target unlabeled samples. That is, if one target sample x_j^u has a high transport probability from one source sample x_i^s (γ_{ij} is high), then it is probable that the predicted label is the same with $y(x_i^s)$.

Based on this idea, we form the group entropic loss to compare the cross entropy between the predicted probability of each target unlabeled sample and the true label of each source sample. It can be written as

$$\mathcal{L}_g = -\frac{1}{n_s}\frac{1}{n_u} \sum_{x_i^s \in X^s} \sum_{x_j^t \in X^u} \gamma_{ij}^* (y(x_i^s) \log \hat{y}(x_j^u)) \tag{7}$$

where $\hat{y}(x_j^u) = \mathcal{F}(\mathcal{G}(x_j^u))$ is the predicted probability of x_j^u. By penalizing couplings with high cross entropies, we can achieve that each unlabeled target sample can be transported from source samples with same class.

3) Intra-domain Feature Discrimination: The center loss is originally proposed to enhance the discriminative power of the deeply learned features for face recognition [54]. Inspired by this, we also hope to ensure that samples belonging to the same class are close to each other in the feature space. Here, we consider the discriminative centroid loss $\mathcal{L}_{cc}$ for the labeled samples in both source domain and target domain.

$$\begin{aligned} \mathcal{L}_{cc} = &\frac{1}{n_s} \sum_{i=1}^{n_s} \|\mathcal{G}(x_i^s) - c_i^s\|_2^2 - (\|c_0^s - c_1^s\|_2^2) \\ &+ \frac{1}{n_l} \sum_{j=1}^{n_l} \|\mathcal{G}(x_j^t) - c_j^t\|_2^2 - (\|c_0^t - c_1^t\|_2^2) \end{aligned} \tag{8}$$

where c_i^s and c_j^t denote the corresponding class center of x_i^s and x_j^t in the source domain and target domain, respectively. We evaluate them by averaging the deep discriminative features of the samples in the corresponding class.

Algorithm 2 Semi-supervised Optimal Transport with Self-paced Ensemble (*SPSSOT*)

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 bins k; 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: Under-sample from l-th bin with $\frac{p_l}{\sum_m p_m} \cdot |\mathcal{P}|$;
11: **end for**
12: Train $SSOT_i$ using newly under-sampled subset according to Algorithm 1;
13: **end for**
14: **return** Final ensemble model $F(\mathcal{D}^s, \mathcal{D}^l, \mathcal{D}^u) = \frac{1}{n}\sum_{m=1}^{n} SSOT_m(\mathcal{D}^s, \mathcal{D}^l, \mathcal{D}^u)$;

$$
\begin{aligned}
c_k^s &= \frac{1}{S_a}\sum_{i=1}^{N_a} \mathcal{G}(x_i^s)\mathcal{I}(y_i^s, k), k \in \{0, 1\} \\
c_k^t &= \frac{1}{S_b}\sum_{j=1}^{N_b} \mathcal{G}(x_j^t)\mathcal{I}(y_j^t, k), k \in \{0, 1\}
\end{aligned}
\tag{9}
$$

where $\mathcal{I}(y_i, k) = \begin{cases} 1, & y_i = k \\ 0, & y_i = 1 - k \end{cases}$, and $S_a = \sum_{i=1}^{N_a} \mathcal{I}(y_i^s, k)$, $S_b = \sum_{j=1}^{N_b} \mathcal{I}(y_j^b, k)$. Ideally, the class centers should be calculated based on all the samples while the procedure is time-consuming. Herein, we compute the class centers by randomly sampling N_a and N_b samples. In our experiments, we set $N_a = \frac{1}{2} \times n_s, N_b = \frac{1}{2} \times n_l$.

4) Training: Here, we introduce the training process of *SSOT*. Considering the three parts of *SSOT*, the training objective can be described as

$$
\min_{\mathcal{G}, \mathcal{F}} \mathcal{L}_{lot} + \lambda \mathcal{L}_g + \beta \mathcal{L}_{cc} \tag{10}
$$

where λ and β denote hyper-parameters that trade-off the contribution of the intra-domain structures and domain alignment, respectively.

The training process is shown in Algorithm 1. Specifically, in each iteration, we use two steps to update the parameters. First, we fix the feature generator $\mathcal{G}$ and the classifier $\mathcal{F}$, and use the optimal transport mechanism (POT (python optimal transport) [39] in our implementation) to calculate the coupling γ (line 6); Second, we fix γ to update $\mathcal{G}$ and $\mathcal{F}$ with the stochastic gradient descent algorithm (line 7).

C. Self-paced Ensemble

When we try to transfer knowledge from the source domain to the target domain, label shift may happen between two domains [18]. That is, the label distribution changes from the source to the target. At the same time, it is ubiquitous that medical diagnostic data is extremely imbalanced. This reveals not only the disproportion between classes but also other difficulties embedded in the nature of data, such as *noises and class overlapping* [55].

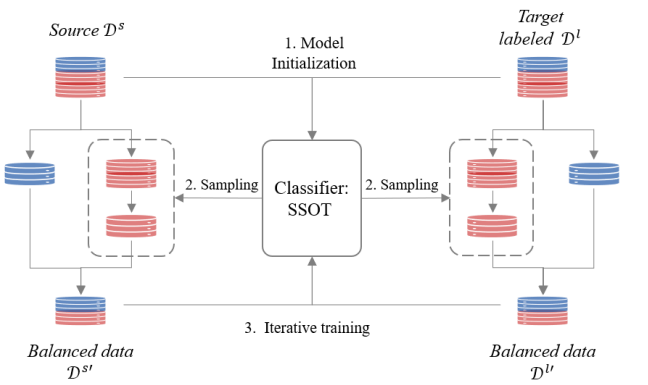


Fig. 3. The Core Idea of Self-paced Ensemble Based on SSOT (SPSSOT). There are 3 main steps: (1) Initialize *SSOT* according to Algorithm. 1; (2) Self-paced under-sampling from majority class in both domains to obtain balanced data; (3) Get an additive model by iteration training. *Best viewed in color.*

Therefore, it is necessary to design a strategy to solve both problems. Inspired by the self-paced ensemble method (SPE) [23], we adapt it to our end-to-end transfer-learning prediction framework so as to sample labeled data simultaneously from two domains. Fig.3 shows the core idea of our SPE-enhanced SSOT (SPSSOT) mechanism. Next, we will present in detail how to carry out the sampling strategy.

1) Classification Hardness Function: We use $\mathcal{H}$ to denote the hardness function, which can be calculated by the summation of individual sample errors, such as absolute error and cross entropy. Given a classifier $\mathcal{F}$ and a sample (x, y), the hardness can be written as $\mathcal{H}(x, y, \mathcal{F}) = |\mathcal{F}(x) - y| \in [0, 1]$. This value contains information that is highly associated with the difficulty of the classification, like noise and model capacity. According to the hardness values, we can divide samples into three types as follows:

- *Trivial samples* account for the largest proportion of samples that are easy to classify, i.e., each of them only contributes tiny hardness. However, the overall contribution is not negligible due to the large number of samples.
- *Noise samples* are different from trivial samples. Though the sample number is small, each sample has a large hardness value. These samples can be caused by the indistinguishable overlapping and will exist stably even when the model is converged.
- *Borderline samples* are the rest training samples.

Intuitively, while sampling, we should (1) keep a small proportion of trivial samples to maintain the original data distribution to avoid overfitting; (2) exclude the interference of noise samples during training; (3) enlarge the weights of borderline samples to improve the model performance. What remains to be settled is how to distinguish three types of samples and achieve under-sampling in practice.

2) Self-paced Under-sampling: There are two important components, *self-paced hardness harmonize* and *combination with SSOT* in achieving self-paced sampling and iterative training. Algorithm 2 describes the detailed process.

Self-paced hardness harmonize. We regard the class with a higher proportion as the *majority* (in Sepsis early detection, the majority class is the patients without Sepsis). After calculating the hardness values of the majority samples, we can split them into k bins regarding different hardness levels, i.e., the l-th bin, B_l, is defined as

$$B_l = \{(x,y)|\frac{l-1}{k} \leq \mathcal{H}(x,y,\mathcal{F}) < \frac{l}{k}\}, \mathcal{H} \in [0,1] \quad (11)$$

Then we can under-sample from every bin by ensuring that the total hardness contribution of each bin is the same, so as to generate a balanced dataset. By harmonizing hardness contribution, the sampling probability of those bins with a larger population will be generally lower. Moreover, we leverage a self-paced factor ω to adjust the decreasing level in training process. This factor is calculated by tan function (line 8 of Algorithm 2) [23]. As ω gets larger, the sampling weights of hard samples will increase. In the beginning, we pay more attention to borderline samples to improve model performance. While in the later iterations (ω becomes very large), the model still keeps a certain number of trivial samples as the "skeleton" to avoid overfitting.

Combination With *SSOT*. Unlike the original SPE [23] that is applied on supervised classification models (e.g., classification models of sklearn[2]), we should solve the data imbalance in two domains and combine the self-paced ensemble strategy with *SSOT*. As shown in Fig.3, we perform self-paced under-sampling simultaneously on source labeled data and target labeled data; then, we obtain two balanced datasets (both source and target domains) for training SSOT iteratively.

V. Experiments

A. Dataset

We conduct our experiments on two widely-used real-life Sepsis detection datasets, **MIMIC-III** [56] and the PhysioNet Computing in Cardiology Challenge 2019 [32] (**Challenge**). Specifically, we extracted the first 48-hour data since patients entered ICUs. As a part of patients' records have a large number of missing values, we screened out the patients whose missing value ratio is less than 80%. To obtain the dynamic change information of the data over a period of time, for every patient, we calculated the maximum, minimum, mean, standard error and latest of each clinical indicator within 6 hours. In this way, a patient's 48-hour ICU stay can be converted to eight 6-hour records (samples). Then, we can use the k-th ($k \in [1,8]$) record to predict whether Sepsis would occur or not in the next 6 hours. Through such preprocessing, we obtain the final data for experiments. Some basic statistic information is enumerated in Table II.

TABLE II
Statistics of the Datasets

	MIMIC	Challenge
# patients	12529	8270
# septic patients	2977	1831
Sepsis prevalence (%)	23.76	22.14
# samples	87501	45674
# samples occur Sepsis in next 6 hours	5032	4869
samples with sepsis (%)	5.75	10.66

B. Compared Algorithms

In our experiments, we split the target data into three parts: 1% as labeled data (we will change the ratio in Sec. V-E.3), 79% as unlabeled data, and 20% as test data. To compare with our method *SPSSOT*, we implement four types of baselines.

- *Source only*: train a classifier only with the source data and directly use it with the target test data.
- *Target only*: train a classifier only with the target labeled data (i.e., 1% of the target data) and use it with the target test data.
- *Source & Target Train Together*: put the source data and the labeled target data together as training data to learn a classifier.
- *Source & Target Transfer*: instead of training together, design specific transfer learning methods to transfer knowledge from the source domain to the target domain.

In the former three types, we all use three classical machine learning algorithms popular in Sepsis early detection, i.e., Logistic Regression (*LR*) [7], Neural Network (*NN*) [9] and *XGBoost* [57]. For the fourth type of baselines, we implement five methods for comparison, including an unsupervised domain adaptation method using transport optimal theory, *DeepJDOT* [40], fine-tuned *NN* (*Finetune*), and three start-of-the-art semi-supervised domain adaptation methods, *MME* [24], *LIRR* [25] and S^3D [26].

C. Experiment Design

There is a self-paced sampling strategy in *SPSSOT* to solve the class imbalance question. For a fair comparison, we also apply the method, SPE [3], to downsample majority data and train ensemble models when using *LR*, *NN* and *XGBoost* as base classifiers, where the hardness function is set to Squared Error, the number of base classifiers is set to 20 and the number of bins is to 15. *LR* and *XGBoost* are trained with default scikit-learn parameters, and *NN* has four linear layers, whose dimension is (256, 128, 128, 2).

In transfer learning methods, we use two linear layers as the feature generator $\mathcal{G}$ and the dimension is (256, 128). The structure of classifier, $\mathcal{F}$, is also two-layer and the dimension is (128, 2), where 2 means binary classification in our task. The batch size is set to 128, the parameter optimization algorithm is SGD, and the learning rate is set to 0.001. In *Finetune*, we first train $\mathcal{G}$ and $\mathcal{F}$ with source data and fine-tune them with target labeled data; both parts are trained for 100 epoches. In

[2]https://scikit-learn.org/stable/

[3]https://github.com/ZhiningLiu1998/self-paced-ensemble

TABLE III
OVERALL EVALUATION RESULTS

	MIMIC → Challenge		Challenge → MIMIC		Average
	AUC	improvement	AUC	improvement	improvement
Source Only					
LR	56.15 ± 0.85	15.94%	72.70 ± 0.35	4.61%	10.28%
NN	59.24 ± 0.75	9.89%	70.53 ± 1.34	7.83%	8.86%
XGBoost	60.81 ± 0.26	7.05%	59.41 ± 0.94	28.01%	17.53%
Target Only					
LR	60.21 ± 0.07	8.12%	71.62 ± 0.50	6.19%	7.16%
NN	60.58 ± 0.14	7.46%	61.92 ± 0.29	22.82%	15.14%
XGBoost	58.90 ± 0.65	10.53%	72.97 ± 0.54	4.10%	7.38%
Source & Target Train Together					
LR	59.90 ± 1.15	8.68%	72.89 ± 0.47	4.34%	6.51%
NN	60.81 ± 0.24	7.05%	71.53 ± 0.09	6.32%	6.69%
XGBoost	60.25 ± 0.35	8.05%	68.71 ± 0.21	10.68%	9.37%
Source & Target Transfer					
DeepJDOT	61.17 ± 0.75	6.42%	72.64 ± 0.39	4.69%	5.56%
Finetune	60.11 ± 0.73	8.30%	71.62 ± 1.72	6.19%	7.25%
MME	61.49 ± 0.84	5.87%	75.07 ± 0.70	1.31%	3.59%
LIRR	62.76 ± 0.95	3.73%	75.35 ± 0.59	0.93%	2.33%
S^3D	61.87 ± 0.61	5.22%	75.56 ± 0.37	0.65%	2.94%
SPSSOT (our)	**65.10 ± 0.24**	-	**76.05 ± 0.54**	-	-

DeepJDOT, we set $\alpha = 0.5$, $\lambda_t = 1.0$, $\lambda_s = 2.0$ and the number of iterations is 5000. In *SPSSOT*, we set $\alpha = 0.05$ in Eq. (4), $\theta_s = 1.0$ in Eq.(4) and $\beta = 0.15$, $\lambda = 0.5$ in Eq. (10). In Algorithm 2, the hardness function is Squared Error, the number of base classifiers is set to 5, the number of bins is to 10, and the number of iterations is set to 5000. In *MME*, *LIRR* and S^3D, we apply the same network structure of $\mathcal{G}$ and $\mathcal{F}$, and keep the same batch size and learning rate. We repeat each experiment for 5 times and record the average results. The parameter sensitivity analysis is conducted later in Sec.V-E.3.

Because over 80% papers about Sepsis prediction reported AUC [11], we also pick it as our performance metric.

Our experiment platform is a server with AMD Ryzen 9 3900X 12-Core Processor, 64 GB RAM and GeForce RTX 3090. We use Python 3.8 with scikit-learn 0.24, POT 0.7 and tensorflow 2.4 on Ubuntu 20.04 for algorithm implementation. Our codes and models can be found on Github[4].

D. Results and Discussion

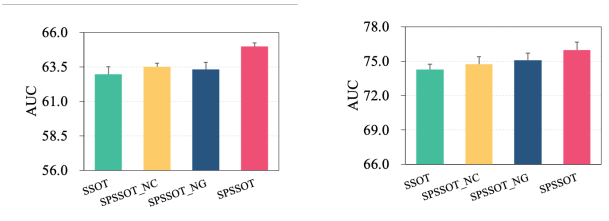
The experiment results of *SPSSOT* and the baselines are reported in Table III. To make a more comprehensive comparison, we demonstrate the experimental results from three perspectives.

First, *SPSSOT* outperforms the other five transfer learning baselines. Between them, *DeepJDOT* is an unsupervised transfer learning method based on Optimal Transport [40]. Thus, the improvement is expected because our *SPSSOT* can further leverage the target labeled data (although the labeled data may be little). Compared to *Finetune*, a common method in transfer learning, the advantage of our method further verifies the effectiveness of using optimal transport to align two feature spaces during the training process. It is worth noting that, while *Finetune* considers 1% labeled data in the target domain, its performance is even worse than *DeepJDOT* without considering any labeled target data. This indicates that even if we have certain labeled data in the target domain, it is still non-trivial to properly leverage the knowledge of such labeled data. In addition, *MME*, *LIRR* and S^3D are the state-of-the-art semi-supervised transfer learning methods. They all outperform the other baselines, which shows that it makes sense to use the labeled data of the source and target domains for domain adaptation at the same time. Specifically, these methods are comparable to *SPSSOT* in Challenge → MIMIC, while *SPSSOT* is over 3% ahead in MIMIC → Challenge. To some degree, this demonstrates that our method can more efficiently use the sparsely labeled target data throughout the knowledge transfer process.

Second, compared to the baseline methods (*LR, NN* and *XGBoost*) that are trained with source data and target labeled data together, *SPSSOT* improves at least 7.05% in MIMIC → Challenge and 4.34% in Challenge → MIMIC. The probable reason is that the feature distributions of two domains are different so that simply putting two domains' data together for training is not effective. When we further compare the models that *train together* and *Finetune*, *Finetune* may not perform better. This result illustrates that though the model trained with source data provides initial parameters for *FineTune*, the initialization probably is not suitable for target data. Therefore, it may appear a negative transfer when the feature distributions of two domains are not similar.

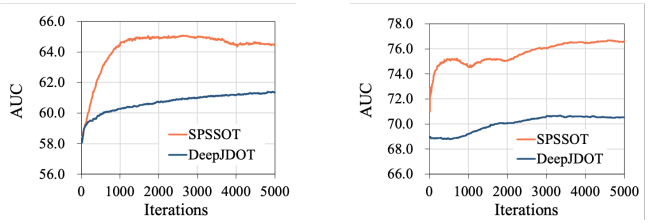
Third, all the no-transfer baselines, i.e., *Source Only* and *Target Only*, perform rather poorly. The results of *Source Only* indicate that, though MIMIC and Challenge both are medical datasets with the same features, there still are some differences. For *Target Only* methods, as we only have a small amount of labeled data (1% in our setting) to train the model, the performance cannot be guaranteed, which is like the *cold start* scenario. Moreover, we can find that the AUC values of *NN* are very small while only using target labeled data, which may be due to the overfitting on a small number of samples.

[4]https://github.com/RuiqingDing/SPSSOT



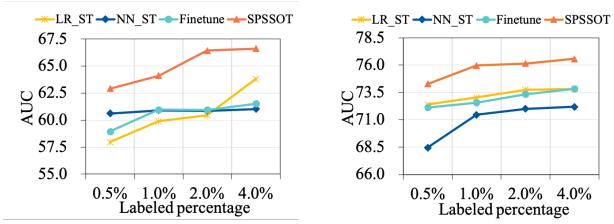
(a) MIMIC → Challenge (b) Challenge → MIMIC

Fig. 4. Ablation Study: compare *SPSSOT* with its variants.



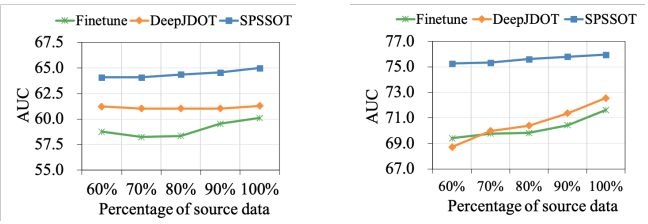
(a) MIMIC → Challenge (b) Challenge → MIMIC

Fig. 5. The convergence performance of *SPSSOT* and *DeepJDOT*.



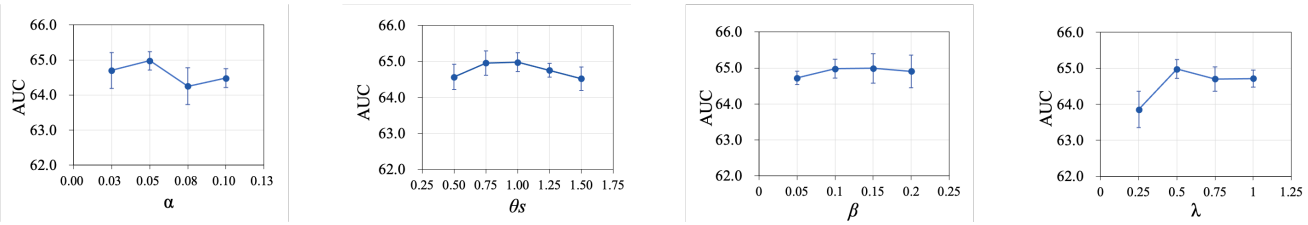
(a) MIMIC → Challenge (b) Challenge → MIMIC

Fig. 6. Different percentages of labeled data in target domain.



(a) MIMIC → Challenge (b) Challenge → MIMIC

Fig. 7. Different sampling percentages of source data.



(a) weight of optimal transport (b) weight of source's classification loss (c) weight of discriminative centroid loss (d) weight of group entropy

Fig. 8. Parameter Sensitivity of MIMIC → Challenge: vary the four hyperparameters in the loss function and compare the results of the experiments.

E. Analysis

1) Ablation Study: To analyse the separate contribution of *SPSSOT*, we compare *SPSSOT* with three variants of *SPSSOT* in this section, as listed below:

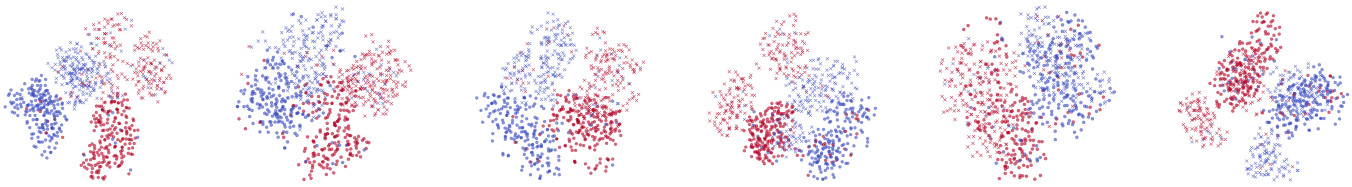
- *SSOT*: we remove Self-paced ensemble from *SPSSOT*.
- *SPSSOT_NC*: we do not consider intra-domain structure during transferring, i.e., $\beta = 0$ in Eq. (10).
- *SPSSOT_NG*: we delete the group entropic loss during training, i.e., $\lambda = 0$ in Eq. (10).

The results are shown in Fig. 4. As we can see, compared with the complete model, *SSOT* is worse. This is because after removing the Self-paced ensemble, the datasets encounter a label imbalance that will result in the difficulty of modeling. *SSPSOT_NC* ignores the intra-domain structure with no consideration of the embedding distances in the hidden feature space; it is thus hard to find a good classification boundary. What's more, *SPSSOT_NG* causes that the paired target unlabeled samples may come from different classes; this would lead to an ambiguous result. In brief, the results indicate that each part of our model *SPSSOT* is necessary.

2) Convergence: To illustrate the convergence of *SPSSOT*, we evaluate the test AUCs of the transfer learning methods, *SPSSOT* and *DeepJDOT*. The results are shown in Fig. 5. It reveals that our model can achieve significantly better test AUCs only with a few iterations and keep relatively stable convergence performance. On the task of Challenge → MIMIC, there are obvious changes in some iterations, like the 1000th and the 2000th iterations. That is because after every 1000 iterations, *SPSSOT* will resample from the majority data and continue training. With the increasing resampling times, the test performance will gradually become stable.

3) Sensitivity Analysis: **Labeled Percentage in Target Domain**. In experiments, we set 1% of the target domain data to have labels by default. To further verify the stability of the method, we adjust the proportion of samples with known labels in the target domain to 0.5%, 2% and 4%. In Fig. 6, we show the results of different target label percentages. It can be observed that *SPSSOT* always performs the best in these percentages. It is reasonable that as the labeled percentage decreases, so does the models' performances. However, compared with the models that are trained with source data and target labeled data, *SPSSOT* keeps a relatively steady trend. This is an ideal situation for practical applications, which means that we can train a transfer learning model with acceptable performance by spending a small amount of cost to label a small amount of data.

Sampling Percentage of Source Data. To validate the effect of different numbers of source domain samples, we conduct experiments on source data with different sample sizes.



(a) Finetune (M→C) (b) DeepJDOT(M→C) (c) SPSSOT (M→C) (d) Finetune (C→M) (e) DeepJDOT(C→M) (f) SPSSOT (C→M)

Fig. 9. Feature visualization. M → C means MIMIC → Challenge and C → M means Challenge → MIMIC. Different colors represent different classes (red: will have Sepsis, blue: will not have Sepsis), different shapes represent different domains (round: source domain, cross: target). *Best viewed in color.*

TABLE IV
RESULTS OF DIFFERENT SEPSIS EARLY DETECTION TARGETS.

Advance Time	MIMIC → Challenge	Challenge → MIMIC
2 hours	63.53 ± 0.51	71.64 ± 0.71
4 hours	64.08 ± 0.35	72.30 ± 0.65
6 hours(default)	65.10 ± 0.24	76.05 ± 0.54

TABLE V
RESULTS OF DIFFERENT FEATURE GENERATORS IN *SPSSOT*.

	MIMIC → Challenge	Challenge → MIMIC
Source Only		
NN	59.24 ± 0.75	70.53 ± 1.34
LSTM	54.58 ± 0.50	68.26 ± 1.01
GRU	54.87 ± 0.97	69.72 ± 0.96
NN+GRU	57.37 ± 0.74	70.53 ± 0.93
Target Only		
NN	60.58 ± 0.14	61.92 ± 0.29
LSTM	54.74 ± 0.93	58.75 ± 1.29
GRU	57.62 ± 0.59	57.29 ± 0.76
NN+GRU	59.35 ± 0.86	59.99 ± 0.54
Source & Target Train Together		
NN	60.81 ± 0.24	71.53 ± 0.09
LSTM	54.67 ± 1.05	68.83 ± 0.97
GRU	58.14 ± 1.10	70.56 ± 0.94
NN+GRU	59.35 ± 0.36	71.52 ± 0.48
Source & Target Transfer		
Finetune (NN)	60.11 ± 0.73	71.62 ± 1.72
Finetune (LSTM)	58.25 ± 0.77	70.03 ± 0.90
Finetune (GRU)	59.60 ± 0.76	70.22 ± 1.00
Finetune (NN+GRU)	60.05 ± 0.49	71.29 ± 1.26
SPSSOT (NN)	**65.10 ± 0.24**	**76.05 ± 0.54**
SPSSOT (LSTM)	60.45 ± 0.82	73.67 ± 0.62
SPSSOT (GRU)	62.09 ± 0.90	73.94 ± 0.50
SPSSOT (NN+GRU)	63.81 ± 0.51	75.18 ± 0.78

Fig. 7 displays the results. We observe that the performance of all methods increases when using more source samples. At the same time, our *SPSSOT* approach consistently outperforms the other two baselines, which demonstrates the effectiveness of our proposed method for knowledge transfer.

Hyper-parameter Sensitivity. There exist four important hyper-parameters in the loss function of *SPSSOT*: the weight of optimal transport α in Eq. (4), the weight of the source data's classification loss θ_s in Eq. (4), the weight of discriminative centroid loss β, and the weight of group entropy λ in Eq. (10). To test the stability of the performances of *SPSSOT*, we take a transfer scenerio, MIMIC → Challenge, as example to test different values of α, θ_s, β and λ. The results are shown in Fig. 8. Comparatively speaking, the model is not sensitive to all these parameters and the AUC just ranges from around 64 to 65. According to the performance, we select the values of these parameters used in our experiments, i.e., $\alpha = 0.05, \theta_s = 1, \beta = 0.15$ and $\lambda = 0.5$.

4) Different Sepsis Early Detection Targets: Early Sepsis detection is potentially life-saving because doctors can treat earlier [32]. In the default setting, we predict Sepsis 6 hours before clinical diagnosis. Here, we add two early detection targets, 2 hours ahead and 4 hours ahead. The results are listed in Table IV. We can find that the performance is best when the advance time is 6 hours. This is because when the advance time is short, the data imbalance will be exacerbated (i.e., the proportion of having Sepsis is decreased).

5) Diverse Feature Generators: Considering the physiological indicators can be regarded as time-series data, we adapt the popular time series networks, LSTM [58] and GRU [59], as the feature generators of *SPSSOT*. As seen in Table V, LSTM and GRU perform worse than NN (the default feature generator). Meanwhile, by fixing the feature generator (e.g., LSTM or GRU), *SPSSOT* consistently performs the best. This verifies the robustness of our method in applying different feature generators. We also concatenate the feature representations of NN and GRU (i.e., NN+GRU), but the results are still not as good as those of using only NN. These results inspire us that for exploiting time series models, perhaps more advanced feature engineering techniques are required.

6) Feature visualization: To show the feature transfer capability, we visualize the t-SNE embeddings [60] of the hidden representation by *Finetune*, *DeepJDOT* and *SPSSOT*. Fig. 9 (a) - 9 (c) correspond to MIMIC → Challenge and Fig. 9 (d) - 9 (f) correspond to Challenge → MIMIC. In each subfigure, different colors denote different categories (red: will have Sepsis, blue: will not have Sepsis), and different shapes denote different domains (round: source domain, cross: target). Fig. 9(a) and Fig. 9(d) display that the features learned by *Finetune* for different domains are almost totally separated, i.e., points represented by different shapes in the same feature space are separated from each other. Fig. 9(b) and Fig. 9(e) illustrate that though the domains can be aligned to a certain extend, the bad thing is some target samples are aligned to the source data with wrong classes, causing negative transfer. Note that, Fig. 9(c) and Fig. 9(f) show that the features generated by *SPSSOT* achieve better domain alignment with a clearer class boundary. Specifically, when comparing Fig. 9(e) and Fig. 9(f), they both can blend the data from two domains well (i.e., the circled points and the crossed points are mixed). However, the classification boundary of *DeepJDOT* is not as clear as that

of *SPSSOT*. In particular, *DeepJDOT*'s blue and red samples near the boundary are more interleaved than *SPSSOT*, which increases the likelihood that they will be incorrectly classified. In a nutshell, the visualization results reveal that our proposal can match the complex structures of the source and target domains as well as maximize the margin between different classes.

VI. Conclusion

In this paper, we describe a new framework based on optimal transport and self-paced ensemble to solve the semi-supervised transfer learning problem for Sepsis early detection in the scenario that there is only little labeled data in the target domain (e.g. hospital). Empirical studies on real-world clinical datasets demonstrate the effectiveness of *SPSSOT* in aligning feature spaces and eliminating the influence of class imbalance. In fact, though *SPSSOT* is proposed for Sepsis early detection, it can be easily adapted for other transfer learning tasks. The only requirement is choosing a suitable structure to extract deep features, e.g., CNNs for image identification [41] and RNNs for time series prediction [58].

It is no doubt that there are still many problems to be solved. First, we only downsample from the labeled data to mitigate the effects of data imbalance. It is worthwhile to think about how to downsample from the target unlabeled data effectively to improve the accuracy of detection. Moreover, we can further explore how to exploit time series models better. After that, when there are private features in the source domain, it is hard to directly apply the optimal transport technique. Because feature similarity cannot be appropriately calculated between a source sample and a target sample. Therefore, it may require incorporating more transfer learning techniques, e.g., knowledge distillation [61]. Finally, privacy protection is an important issue that cannot be ignored when models are implemented in real-world applications. However, the constraints of privacy protection clauses often prevent data from being moved to the data center for unified storage and training. Federated learning [62] provides a new idea and still needs to be explored.

References

[1] M. Singer, C. S. Deutschman, C. W. Seymour, M. Shankar-Hari, D. Annane, M. Bauer, R. Bellomo, G. R. Bernard, J.-D. Chiche, C. M. Coopersmith, R. S. Hotchkiss, M. M. Levy, J. C. Marshall, G. S. Martin, S. M. Opal, G. D. Rubenfeld, T. van der Poll, J.-L. Vincent, and D. C. Angus, "The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)," *JAMA*, vol. 315, no. 8, pp. 801–810, 02 2016.

[2] J. J. Zimmerman, "Pediatric sepsis from start to finish," *Pediatric critical care medicine: a journal of the Society of Critical Care Medicine and the World Federation of Pediatric Intensive and Critical Care Societies*, vol. 16, no. 5, p. 479, 2015.

[3] C. Fleischmann, A. Scherag, N. K. Adhikari, C. S. Hartog, T. Tsaganos, P. Schlattmann, D. C. Angus, and K. Reinhart, "Assessment of global incidence and mortality of hospital-treated sepsis. current estimates and limitations," *American journal of respiratory and critical care medicine*, vol. 193, no. 3, pp. 259–272, 2016.

[4] L. Ou, J. Chen, K. Hillman, A. Flabouris, M. Parr, H. Assareh, and R. Bellomo, "The impact of post-operative sepsis on mortality after hospital discharge among elective surgical patients: a population-based cohort study," *Critical Care*, vol. 21, no. 1, pp. 1–13, 2017.

[5] E. Sheetrit, N. Nissim, D. Klimov, L. Fuchs, Y. Elovici, and Y. Shahar, "Temporal pattern discovery for accurate sepsis diagnosis in icu patients," *arXiv:1709.01720*, 2017.

[6] A. Kumar, D. Roberts, K. E. Wood, B. Light, J. E. Parrillo, S. Sharma, R. Suppes, D. Feinstein, S. Zanotti, L. Taiberg *et al.*, "Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock," *Critical care medicine*, vol. 34, no. 6, pp. 1589–1596, 2006.

[7] S. P. Shashikumar, M. D. Stanley, I. Sadiq, Q. Li, A. Holder, G. D. Clifford, and S. Nemati, "Early sepsis detection in critical care patients using multiscale blood pressure and heart rate dynamics," *Journal of electrocardiology*, vol. 50, no. 6, pp. 739–743, 2017.

[8] S. Horng, D. A. Sontag, Y. Halpern, Y. Jernite, N. I. Shapiro, and L. A. Nathanson, "Creating an automated trigger for sepsis clinical decision support at emergency department triage using machine learning," *PloS one*, vol. 12, no. 4, p. e0174708, 2017.

[9] J. Futoma, S. Hariharan, and K. Heller, "Learning to detect sepsis with a multitask gaussian process rnn classifier," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1174–1182.

[10] X. Li, X. Xu, F. Xie, X. Xu, Y. Sun, X. Liu, X. Jia, Y. Kang, L. Xie, F. Wang *et al.*, "A time-phased machine learning model for real-time prediction of sepsis in critical care," *Critical Care Medicine*, vol. 48, no. 10, pp. e884–e888, 2020.

[11] L. M. Fleuren, T. L. Klausch, C. L. Zwager, L. J. Schoonmade, T. Guo, L. F. Roggeveen, E. L. Swart, A. R. Girbes, P. Thoral, A. Ercole *et al.*, "Machine learning for the prediction of sepsis: a systematic review and meta-analysis of diagnostic test accuracy," *Intensive care medicine*, vol. 46, no. 3, pp. 383–400, 2020.

[12] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[13] G. Lee, I. Rubinfeld, and Z. Syed, "Adapting surgical models to individual hospitals using transfer learning," in *2012 IEEE 12th international conference on data mining workshops*. IEEE, 2012, pp. 57–63.

[14] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor ai: Predicting clinical events via recurrent neural networks," in *Machine learning for healthcare conference*. PMLR, 2016, pp. 301–318.

[15] P. Gupta, P. Malhotra, J. Narwariya, L. Vig, and G. Shroff, "Transfer learning for clinical time series analysis using deep neural networks," *Journal of Healthcare Informatics Research*, vol. 4, no. 2, pp. 112–137, 2020.

[16] H. S. Luft, D. W. Garnick, D. H. Mark, D. J. Peltzman, C. S. Phibbs, E. Lichtenberg, and S. J. McPhee, "Does quality influence choice of hospital?" *Jama*, vol. 263, no. 21, pp. 2899–2906, 1990.

[17] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2010.

[18] K. Azizzadenesheli, A. Liu, F. Yang, and A. Anandkumar, "Regularized learning for domain adaptation under label shifts." in *ICLR*, 2019.

[19] C. Cortes, Y. Mansour, and M. Mohri, "Learning bounds for importance weighting." in *Nips*, vol. 10. Citeseer, 2010, pp. 442–450.

[20] Y. Yang and Z. Xu, "Rethinking the value of labels for improving class-imbalanced learning," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[21] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[22] C. Villani, *Optimal transport: old and new*. Springer Science & Business Media, 2008, vol. 338.

[23] Z. Liu, W. Cao, Z. Gao, J. Bian, H. Chen, Y. Chang, and T.-Y. Liu, "Self-paced ensemble for highly imbalanced massive data classification," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 2020, pp. 841–852.

[24] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," *ICCV*, 2019.

[25] B. Li, Y. Wang, S. Zhang, D. Li, K. Keutzer, T. Darrell, and H. Zhao, "Learning invariant representations and risks for semi-supervised domain adaptation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1104–1113.

[26] J. Yoon, D. Kang, and M. Cho, "Semi-supervised domain adaptation via sample-to-sample self-distillation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2022, pp. 1978–1987.

[27] G. Monge, "Mémoire sur la théorie des déblais et des remblais," *Histoire de l'Académie Royale des Sciences de Paris*, 1781.

[28] A. Karpatne, I. Ebert-Uphoff, S. Ravela, H. A. Babaie, and V. Kumar, "Machine learning for the geosciences: Challenges and opportunities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 8, pp. 1544–1554, 2018.

[29] R. Joshi, D. Kommers, L. Oosterwijk, L. Feijs, C. van Pul, and P. Andriessen, "Predicting neonatal sepsis using features of heart rate variability, respiratory characteristics, and ecg-derived estimates of infant motion," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 3, pp. 681–692, 2020.

[30] C. León, G. Carrault, P. Pladys, and A. Beuchée, "Early detection of late onset sepsis in premature infants using visibility graph analysis of heart rate variability," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 4, pp. 1006–1017, 2021.

[31] F. van Wyk, A. Khojandi, and R. Kamaleswaran, "Improving prediction performance using hierarchical analysis of real-time data: A sepsis case study," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 3, pp. 978–986, 2019.

[32] M. A. Reyna, C. Josef, S. Seyedi, R. Jeter, S. P. Shashikumar, M. B. Westover, A. Sharma, S. Nemati, and G. D. Clifford, "Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019," in *2019 Computing in Cardiology (CinC)*. IEEE, 2019, pp. Page–1.

[33] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 9, pp. 1853–1865, 2016.

[34] N. Courty, R. Flamary, and D. Tuia, "Domain adaptation with regularized optimal transport," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2014, pp. 274–289.

[35] M. Perrot, N. Courty, R. Flamary, and A. Habrard, "Mapping estimation for discrete optimal transport," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 4204–4212.

[36] I. Redko, A. Habrard, and M. Sebban, "Theoretical analysis of domain adaptation with optimal transport," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2017, pp. 737–753.

[37] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy, "Joint distribution optimal transportation for domain adaptation," in *NIPS*, 2017.

[38] Y. Yan, W. Li, H. Wu, H. Min, M. Tan, and Q. Wu, "Semi-supervised optimal transport for heterogeneous domain adaptation." in *IJCAI*, vol. 7, 2018, pp. 2969–2975.

[39] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, and T. Vayer, "Pot: Python optimal transport," *Journal of Machine Learning Research*, vol. 22, no. 78, pp. 1–8, 2021.

[40] B. B. Damodaran, B. Kellenberger, R. Flamary, D. Tuia, and N. Courty, "Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 447–463.

[41] R. Xu, P. Liu, L. Wang, C. Chen, and J. Wang, "Reliable weighted optimal transport for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4394–4403.

[42] C. Elkan, "The foundations of cost-sensitive learning," in *International joint conference on artificial intelligence*, vol. 17, no. 1. Lawrence Erlbaum Associates Ltd, 2001, pp. 973–978.

[43] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "Smoteboost: Improving prediction of the minority class in boosting," in *European conference on principles of data mining and knowledge discovery*. Springer, 2003, pp. 107–119.

[44] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[45] Y. Freund, R. E. Schapire *et al.*, "Experiments with a new boosting algorithm," in *icml*, vol. 96. Citeseer, 1996, pp. 148–156.

[46] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds., 2010, pp. 1189–1197.

[47] D. Zhang, J. Han, L. Zhao, and D. Meng, "Leveraging prior-knowledge for weakly supervised object detection under a collaborative self-paced curriculum learning framework," *Int. J. Comput. Vis.*, vol. 127, no. 4, pp. 363–380, 2019.

[48] D. Zhang, D. Meng, and J. Han, "Co-saliency detection via a self-paced multiple-instance learning framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 865–878, 2017.

[49] D. Zhang, J. Han, L. Yang, and D. Xu, "SPFTN: A joint learning framework for localizing and segmenting objects in weakly labeled videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 475–489, 2020.

[50] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[51] C. W. Seymour, V. X. Liu, T. J. Iwashyna, F. M. Brunkhorst, T. D. Rea, A. Scherag, G. Rubenfeld, J. M. Kahn, M. Shankar-Hari, M. Singer, C. S. Deutschman, G. J. Escobar, and D. C. Angus, "Assessment of Clinical Criteria for Sepsis: For the Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)," *JAMA*, vol. 315, no. 8, pp. 762–774, 02 2016.

[52] M. Yang, C. Liu, X. Wang, Y. Li, H. Gao, X. Liu, and J. Li, "An explainable artificial intelligence predictor for early detection of sepsis," *Critical Care Medicine*, vol. 48, no. 11, pp. e1091–e1096, 2020.

[53] S. Angenent, S. Haker, and A. Tannenbaum, "Minimizing flows for the monge–kantorovich problem," *SIAM journal on mathematical analysis*, vol. 35, no. 1, pp. 61–97, 2003.

[54] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*. Springer, 2016, pp. 499–515.

[55] D. Gamberger, N. Lavrac, and C. Groselj, "Experiments with noise filtering in a medical domain," in *ICML*, vol. 99, 1999, pp. 143–151.

[56] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[57] M. Zabihi, S. Kiranyaz, and M. Gabbouj, "Sepsis prediction in intensive care unit using ensemble of xgboost models," in *2019 Computing in Cardiology (CinC)*. IEEE, 2019, pp. Page–1.

[58] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[59] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv:1412.3555*, 2014.

[60] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[61] L. Ma, X. Ma, J. Gao, X. Jiao, Z. Yu, C. Zhang, W. Ruan, Y. Wang, W. Tang, and J. Wang, "Distilling knowledge from publicly available online emr data to emerging epidemic for prognosis," in *Proceedings of the Web Conference 2021*, 2021, pp. 3558–3568.

[62] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[63] V. L. Parsons, "Stratified sampling," *Wiley StatsRef: Statistics Reference Online*, pp. 1–11, 2014.

[64] F. T. Liu, K. M. Ting, and Z. Zhou, "Isolation forest," in *Proceedings of the 8th IEEE International Conference on Data Mining, pages = 413–422, publisher = IEEE Computer Society, year = 2008,*.

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 filter out the shared features of two domains (listed in Table I) and adopt a domain-shared feature generator $\mathcal{G}$. However, both datasets have their own private features, which are enumerated 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).

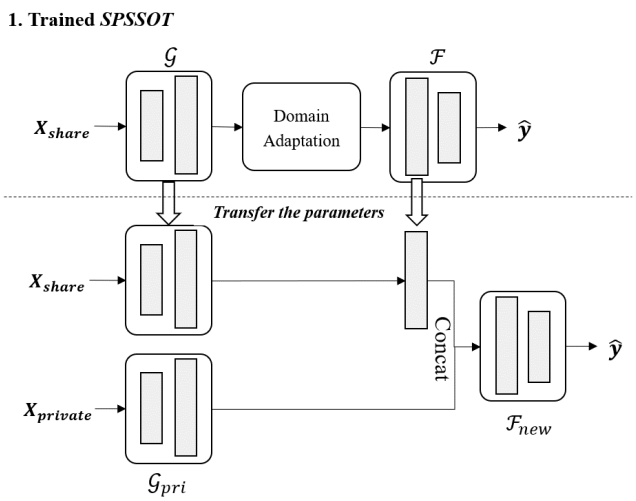


Fig. 10. The network structure to transfer *SPSSOT*'s parameters to target domain with private features. $\boldsymbol{X_{share}}$ means only using shared features as the input, similarly, $\boldsymbol{X_{private}}$ means only using target private features as input.

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].