Adaptive Multi-Source Causal Inference from Observational Data

Thanh Vinh Vo
National University of Singapore
votv@comp.nus.edu.sg

Pengfei Wei
AI Lab Speech & Audio Bytedance
Singapore
wpf89928@gmail.com

Trong Nghia Hoang
Washington State University
trongnghia.hoang@wsu.edu

Tze-Yun Leong
National University of Singapore
leongty@comp.nus.edu.sg

ABSTRACT

We propose a new approach to estimate causal effects from observational data. We leverage multiple data sources which share similar causal mechanisms with the scarce target observations to help infer causal effects in the target domain. The data sources may be available in sequence or some unplanned order. Causal inference can be carried out without prior knowledge of the data discrepancy between the source and target observations. We introduce three levels of knowledge transfer through modelling the outcomes, treatments, and confounders to achieve consistent positive transfer. We incorporate parametric transfer factors to adaptively control the transfer strength, thus achieving a fair and balanced knowledge transfer between the sources and the target. We also empirically show the effectiveness of the proposed method as compared with recent baselines.

CCS CONCEPTS

• **Computing methodologies → Causal reasoning and diagnostics**; **Transfer learning**; *Kernel methods*; *Latent variable models*; • **Mathematics of computing → Causal networks**; *Variational methods*.

KEYWORDS

causal inference, heterogeneous treatment effect, transfer learning, representer theorem

ACM Reference Format:
Thanh Vinh Vo, Pengfei Wei, Trong Nghia Hoang, and Tze-Yun Leong. 2022. Adaptive Multi-Source Causal Inference from Observational Data. In *Proceedings of the 31st ACM Int'l Conference on Information and Knowledge Management (CIKM '22), Oct. 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3511808.3557230

1 INTRODUCTION

Causal inference for estimating treatment effects of an intervention on a particular outcome commonly arises in many practical areas, e.g., personalized medicine [16, 37], digital experiments [48] and political science [13]. This process is complicated by the presence of latent confounders that affect both the treatment and the outcome [24, 26, 38], e.g., a patient's socioeconomic status, which cannot be directly observed, affects both affordable therapy options (the treatment) and the health conditions (the outcome) of this patient. Treatment effects with latent confounders are usually estimated using a set of proxy variables with sufficient data observations, e.g., using income, residential address, etc., as proxies for socioeconomic status. However, the observational data of a population may be scarce in practice, possibly due to difficulty in collection or expensive annotation, leading to poor estimates of the treatment effects in that population. This problem is exacerbated in cases with heterogeneous treatment effects, which means that *the same treatment may affect different individuals or populations differently* [14, 18].

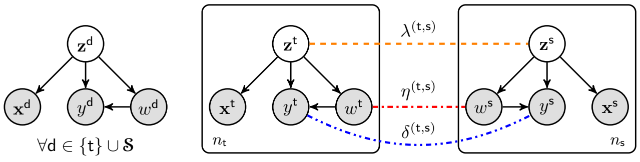
Fortunately, observations from experiments of the same treatment on different populations are likely to share similar causal mechanisms, e.g., causal graphs and structural causal equations. Directly combining the data from the source and target populations, however, might give a better *global* causal estimand, but not the *heterogeneous treatment effects*. How to adaptively transfer useful knowledge from the source to the target is a challenging problem.

For example, suppose we have sufficient observational data in a region A to estimate the treatment effects of a new medicine (the treatment) on blood pressure (the outcome) of patients in that region. We wish to utilize the data of (source) region A to help infer causal effects in another (target) region B whose data is scarce. However, the average population age in region A may be different from that in region B. The distribution of blood pressure for different age groups may also be different [41], i.e., the distribution of the outcomes in A and B are different. If we naively combine the two datasets, the population with more data (A) might dominate the one with less (B), leading to a biased causal effect estimation for region B, especially for heterogeneous treatment effects. In such cases, matching methods such as covariate or propensity score matching may result in a very small dataset, which might reduce the accuracy of the estimated heterogeneous treatment effects in the target population. Furthermore, the feature "age" might not even be available in the datasets, which could further bias the estimation. Other problematic situations include those where the distributions of the other features or covariates in the system are also different. We further illustrate this bias through an example in Appendix A.

In this work, we examine how to improve treatment effect estimation on a target population by exploiting useful information from some different but related data sources, taking into account the potential distribution dissimilarities between the source and target populations. Our contributions are summarized as follows:

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10.
https://doi.org/10.1145/3511808.3557230


$\mathbf{z}^{\mathrm{d}}$
$\mathbf{x}^{\mathrm{d}}$
y^{d}
w^{d}
$\forall \mathrm{d} \in \{\mathrm{t}\} \cup \mathcal{S}$
$\mathbf{z}^{\mathrm{t}}$
$\lambda^{(\mathrm{t,s})}$
$\mathbf{z}^{\mathrm{s}}$
$\mathbf{x}^{\mathrm{t}}$
y^{t}
w^{t}
$\eta^{(\mathrm{t,s})}$
w^{s}
y^{s}
$\mathbf{x}^{\mathrm{s}}$
$\delta^{(\mathrm{t,s})}$
n_{t}
n_{s}

(a) (b)

Figure 1: (a) The causal graph of population d**, where** d **can be either a source population or the target population, i.e.,** $\mathrm{d} \in \{\mathrm{t}\} \cup \mathcal{S}$**, where** $\mathcal{S} = \{\mathrm{s}_1, \mathrm{s}_2, ..., \mathrm{s}_m\}$ **denotes the set of** m **sources.** w^{d}, y^{d}, x^{d}, **and** z^{d} **are the treatment, outcome, proxy variable, and latent confounder, respectively. Further details of these variables are explained in Section 3.2. (b) The three levels of knowledge transfer in our learning algorithm. The dashed lines indicate where the transfer learning happens in the inference, and they do not indicate causal relationships.**

- We introduce the AdaTRANS[1] (adaptive transfer) causal effect estimator that can *adaptively exploit observations* from multiple source populations to help infer the *heterogeneous treatment effects* in a target population with scarce data.
- AdaTRANS can infer causal effects in the target population by utilizing multiple data sources without prior knowledge of data discrepancy between the source and target populations. AdaTRANS can learn the discrepancy between the target population and each of the source populations, and then transfer useful knowledge from the source populations to the target population, overcoming the distribution dissimilarity problem.
- We develop three levels of knowledge transfer in the inference of the outcome, treatment, and confounders. These three levels of knowledge transfer are controlled by three sets of similarity coefficients learned from the observed sources and target data. Specifically, we focus on a causal graph as shown in Figure 1(a). Figure 1(b) is an illustration with one source population that helps to estimate causal effects in the target population. The three similarity coefficients $\lambda^{(\mathrm{t,s})}$, $\nu^{(\mathrm{t,s})}$, $\delta^{(\mathrm{t,s})}$ are learned from the observed data.
- To learn the model, we propose an augmented representer theorem-based variational inference procedure to approximate the posteriors of the confounding factors, which leads to efficient estimation of the treatment effects.
- AdaTRANS is empirically shown to outperform the baselines on data with dissimilar distributions between the sources and the target.

2 RELATED WORK

Causal inference without transfer. A confounder in causal inference may induce bias in the estimates of treatment effects. Classical methods that deal with confounders such as covariate matching, propensity score matching, Bayesian imputations for missing data [43–45] are based on *the ignorability assumption* in the potential outcomes framework to estimate the causal effects from observational data. Modern causal effect estimators for heterogeneous treatment effects including [2, 10, 15, 17, 21, 30–32, 46, 47, 51, 53, 54, 56] are also based on the ignorability assumption in the potential outcomes framework with observed confounders. The central idea is to build a modern regression model to predict the counterfactual outcomes, and then use them to compute the treatment effects of interest, e.g., individual treatment effect (also known as conditional average treatment effect). Veitch et al. [50] proposed a propensity score matching with random mini-batch data. Some other efforts take into account the unobserved confounders in causal inference: [9, 19, 22, 24–26, 29, 40, 52]. Specifically, some proxy variables are introduced to infer the latent confounders. These methods focus only on causal inference in single population, while our work considers *transfer learning* from multiple data sources to estimate causal effects in a target population.

Causal inference with transfer. A line of closely related work: Bareinboim et al. [5], Bareinboim and Pearl [6, 7, 8], Lee et al. [23], Pearl and Bareinboim [36] formalizes the notion of *transportability* of interventions on the source populations to compute causal effects in the target population. The source population can obtain interventional data by conducting randomized trials while only observational data are available in the target population. Aglietti et al. [1], for example, built a joint model based on interventional source data and observational target data. Transportability allows us to use observational data (instead of randomized trials) of the source population to transport to the target population if we can use the *do*-calculus of Pearl [34] to reduce interventional distributions on the source populations to an expression containing only conditional distributions.

We focus on estimating heterogenous treatment effects with latent confounders from only observational data on both the source and target populations, i.e., no randomised experiments are conducted to collect the data. Our method incorporates the idea of transfer learning to learn the discrepancy (or similarity) of each source population and the target population, and then estimates treatment effects in the target population. This is also different from existing work that utilizes tools in causality to improve transfer learning algorithms [e.g., 27, 39, 42, 49, 55].

3 THE PROPOSED METHOD

This section develops a novel causal inference framework that is capable of adaptively exploiting additional data sources to help estimate treatment effects in a target population. Based on the structural causal model (SCM) [35], the aim of our approach is to use additional but related data sources for more accurate inference. We assume that the source observations are related with target ones in the sense that they have the identical causal graph (Figure 1) and structural causal equations (presented in Section 3.2). However, the data distributions may differ considerably across populations, which renders the failure of the straightforward data fusion[2]. To achieve the *positive* and *adaptive* knowledge transfer, instead of modelling those probabilities for each population separately, we propose an augmented-representer theorem with a similarity measurement controlling knowledge transfer strength from the sources to the target population.

Specifically, we develop three levels of knowledge transfer, which take place in learning distributions of the outcome, treatment and confounder. For each level of knowledge transfer, we assign a set

[1] Source code: `https://github.com/vothanhvinh/AdaTRANS`

[2] As shown in [33], a brute-force data fusion of different populations may result in negative knowledge transfer.

of learnable coefficients to model similarity of the corresponding data observations between each pair of populations, and hence adaptively transfer knowledge from multiple source populations to the target population for estimating causal effects of the target population.

3.1 Problem Description

In this work, we focus on one target population t and m source populations $\mathrm{s}_1, \mathrm{s}_2, ..., \mathrm{s}_m$. We denote the set of all source populations as $\boldsymbol{\mathcal{S}} = \{\mathrm{s}_i\}_{i=1}^{m}$. For each population $\mathrm{d} \in \{\mathrm{t}\} \cup \boldsymbol{\mathcal{S}}$, we assume that a finite collection of training data tuples $\{(y_i^{\mathrm{d}}, w_i^{\mathrm{d}}, \mathbf{x}_i^{\mathrm{d}})\}_{i=1}^{n_{\mathrm{d}}}$ is provided. Here, w_i^{d}, y_i^{d} and $\mathbf{x}_i^{\mathrm{d}}$ denote the observed data of the treatment, the outcome and the proxy variable of individual i in the population d, respectively. The source and target populations share the same causal graph as shown in Figure 1(a), but may be different in their structural equations. Hence, their data distributions may be different, e.g., $p_{\mathrm{s}}(\mathbf{x}_i^{\mathrm{d}}, w_i^{\mathrm{d}}, y_i^{\mathrm{d}}) \neq p_{\mathrm{t}}(\mathbf{x}_i^{\mathrm{d}}, w_i^{\mathrm{d}}, y_i^{\mathrm{d}})$, where $p_{\mathrm{s}}(\cdot)$ and $p_{\mathrm{t}}(\cdot)$ denote the distributions of a source s and the target t, respectively. As a result, their causal effects might also be different. Moreover, the target population has scarce training observations while the source populations have sufficient training observations, $n_{\mathrm{t}} \ll \sum_{\mathrm{s} \in \boldsymbol{\mathcal{S}}} n_{\mathrm{s}}$. The objective is to estimate causal effects on a set of individuals of the target population t by utilizing the training observations from t and all source populations $\mathrm{s} \in \boldsymbol{\mathcal{S}}$. In particular, we first train a model that utilizes training data from the source populations and the target population. Then, we use this learned model to estimate individual treatment effect (ITE) and average treatment effect (ATE) in a set of *new* individuals of the target population whose observed proxy variables are $\{\mathbf{x}_{*i}^{\mathrm{t}}\}_{i=1}^{n_*}$, where n_* is the size of this set. These quantities are defined as follows.

Definition 1. *Let Y, W, X be random variables of the outcome, treatment, and proxy variable, respectively. Then, the* ITE *and* ATE *are defined as follows*

$$\begin{aligned}\mathtt{ite}(x) &:= E\big[Y|\mathrm{do}(W{=}1), X{=}x\big] - E\big[Y|\mathrm{do}(W{=}0), X{=}x\big],\\ \mathtt{ate} &:= E[\mathtt{ite}(X)],\end{aligned}$$

where $\mathrm{do}(W{=}w)$ *represents that a treatment* $w \in \{0, 1\}$ *is given to the individual.*

The ITE defined here is also known as the conditional average treatment effect (CATE) [24, 26]. From Definition 1, the ITE and ATE in the above set of individuals of the target population are obtained by $\mathtt{ite}(\mathbf{x}_{*i}^{\mathrm{t}})$ and $\mathtt{ate} = \sum_{i=1}^{n_*} \mathtt{ite}(\mathbf{x}_{*i}^{\mathrm{t}})/n_*$, respectively.

3.2 The Structural Causal Equations

To estimate treatment effects, we first specify the structural equations associated with the causal graph in Figure 1(a). For each $\mathrm{d} \in \{\mathrm{t}\} \cup \boldsymbol{\mathcal{S}}$, we assume the following components.

The latent confounder $\mathbf{z}_i^{\mathbf{d}}$. In real world applications, it is not possible to capture all the potential confounders as some of them might not be observed due to lack of measurement methods or unknown confounders. With the existence of latent confounders, causal inference can lead to a biased estimation. The increasing availability of large and rich datasets enables unobserved confounders to be inferred from other observed variables which are known as the proxy variables. We assume the structural equation of $\mathbf{z}_i^{\mathrm{d}}$ as follows

$$\mathbf{z}_i^{\mathrm{d}} = \boldsymbol{\mu} + \boldsymbol{e}_i^{\mathrm{d}}, \tag{1}$$

where $\boldsymbol{e}_i^{\mathrm{d}} \sim \mathrm{N}(\mathbf{0}, \sigma_z^2 \mathbf{I})$ is the noise and $\boldsymbol{\mu}$ is the mean vector of d_z dimensions.

The outcome $y_i^{\mathbf{d}}$. In practice, the outcome can take different values, such as a binary value or a real number, depending on the the nature of data and the application. We model two cases of the outcome by the following structural equations:

$$\text{Continuous outcome:} \qquad y_i^{\mathrm{d}} = f_y\left(w_i^{\mathrm{d}}, \mathbf{z}_i^{\mathrm{d}}\right) + o_i^{\mathrm{d}}, \tag{2}$$

$$\text{Binary outcome:} \qquad y_i^{\mathrm{d}} = \mathbb{1}\left(o_i^{\mathrm{d}} \leq \varphi\left(f_y\left(w_i^{\mathrm{d}}, \mathbf{z}_i^{\mathrm{d}}\right)\right)\right). \tag{3}$$

In case of continuous outcomes, $o_i^{\mathrm{d}} \sim \mathrm{N}(0, \sigma_y^2)$, where σ_y^2 is the variance. In case of binary outcomes, $o_i^{\mathrm{d}} \sim \mathrm{U}[0, 1]$, where $\varphi(\cdot)$ is the logistic function and $\mathbb{1}(\cdot)$ is the indicator function. In this case, Eq. (3) implies that y_i^{d} (given w_i^{d} and $\mathbf{z}_i^{\mathrm{d}}$) follows Bernoulli distribution where $\varphi\left(f_y\left(w_i^{\mathrm{d}}, \mathbf{z}_i^{\mathrm{d}}\right)\right)$ denotes the probability that $y_i^{\mathrm{d}} = 1$. For both cases, the function $f_y(\cdot)$ is modelled in the following form: $f_y(w_i^{\mathrm{d}}, \mathbf{z}_i^{\mathrm{d}}) = w_i^{\mathrm{d}} f_{y_1}(\mathbf{z}_i^{\mathrm{d}}) + (1 - w_i^{\mathrm{d}}) f_{y_0}(\mathbf{z}_i^{\mathrm{d}})$, where $f_{y_1} : \mathcal{Z} \mapsto \mathcal{F}_{y_1}$ and $f_{y_0}^{\mathrm{d}} : \mathcal{Z} \mapsto \mathcal{F}_{y_0}$ are functions modelling the outcome when $w_i^{\mathrm{d}} = 1$ and $w_i^{\mathrm{d}} = 0$, respectively. $\mathcal{Z}$ is the set containing $\mathbf{z}_i^{\mathrm{d}}$. $\mathcal{F}_{y_1}$ and $\mathcal{F}_{y_0}$ are Hilbert spaces.

The treatment $w_i^{\mathbf{d}}$ and the proxy variable $\mathbf{x}_i^{\mathbf{d}}$. Similar to the outcome, we specify

$$w_i^{\mathrm{d}} = \mathbb{1}\left(u_i^{\mathrm{d}} \leq \varphi\left(f_w(\mathbf{z}_i^{\mathrm{d}})\right)\right), \tag{4}$$

$$x_{ik}^{\mathrm{d}} = f_x(\mathbf{z}_i^{\mathrm{d}})_k + r_{ik}^{\mathrm{d}} \qquad \text{for continuous } x_{ik}^{\mathrm{d}}, \tag{5}$$

$$x_{ik}^{\mathrm{d}} = \mathbb{1}\left(r_{ik}^{\mathrm{d}} \leq \varphi(f_x(\mathbf{z}_i^{\mathrm{d}})_k)\right) \qquad \text{for binary } x_{ik}^{\mathrm{d}}, \tag{6}$$

where $f_w : \mathcal{Z} \mapsto \mathcal{F}_w$ and $f_x : \mathcal{Z} \mapsto \mathcal{F}_x$ are functions, $\mathcal{F}_w$ and $\mathcal{F}_x$ are Hilbert spaces. In Eq. (5) and (6), $f_x(\mathbf{z}_i^{\mathrm{d}})_k$ denotes the k-th dimension of $f_x(\mathbf{z}_i^{\mathrm{d}})$, $r_{ik}^{\mathrm{d}} \sim \mathrm{N}(0, (\sigma_{xk}^{\mathrm{d}})^2)$ for continuous x_{ik}^{d} and $r_{ik}^{\mathrm{d}} \sim \mathrm{U}[0, 1]$ for binary x_{ik}^{d}. Finally, $u_i^{\mathrm{d}} \sim \mathrm{U}[0, 1]$ in Eq. (4).

In the subsequent section, we develop an augmented-representer theorem algorithm to learn the functions f_c (where $c \in \{y_0, y_1, x, w\}$) such that it can adaptively transfer knowledge from the sources to the target population. We then estimate causal effects in the target population based on these learned functions.

3.3 Estimating Treatment Effects

Definition 1 implies that the central task to estimate ITE and ATE in the target population is to find $p(y_i^{\mathrm{t}}|\mathrm{do}(w_i^{\mathrm{t}}), \mathbf{x}_i^{\mathrm{t}})$. With existence of the latent confounder $\mathbf{z}_i^{\mathrm{t}}$, we can further expand this quantity using the backdoor adjustment formula [34] as follows

$$p(y_i^{\mathrm{t}}|\mathrm{do}(w_i^{\mathrm{t}}), \mathbf{x}_i^{\mathrm{t}}) = \int p(y_i^{\mathrm{t}}|w_i^{\mathrm{t}}, \mathbf{z}_i^{\mathrm{t}}) p(\mathbf{z}_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}}) d\mathbf{z}_i^{\mathrm{t}}. \tag{7}$$

The above equation shows that the causal effect is identifiable if we can find the conditional distributions $p(y_i^{\mathrm{t}}|w_i^{\mathrm{t}}, \mathbf{z}_i^{\mathrm{t}})$ and $p(\mathbf{z}_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}})$. The second distribution can be further expanded by $p(\mathbf{z}_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}}) = \sum_{w_i^{\mathrm{t}}} \int p(\mathbf{z}|\mathbf{x}_i^{\mathrm{t}}, y_i^{\mathrm{t}}, w_i^{\mathrm{t}}) p(y_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}}, w_i^{\mathrm{t}}) p(w_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}}) \mathrm{d}y_i^{\mathrm{t}}$. Following the forward sampling strategy, the remaining task is to find the following distributions

$$p(w_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}}), \quad p(y_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}}, w_i^{\mathrm{t}}), \quad p(\mathbf{z}_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}}, y_i^{\mathrm{t}}, w_i^{\mathrm{t}}), \quad p(y_i^{\mathrm{t}}|w_i^{\mathrm{t}}, \mathbf{z}_i^{\mathrm{t}}), \tag{8}$$

and then orderly draw samples from these estimated distributions to obtain the empirical expectation of y_i^{t} given $\mathrm{do}(w_i^{\mathrm{t}})$ and $\mathbf{x}_i^{\mathrm{t}}$. Due to the data scarcity issue in the target population t, the estimations using target observations only may not be accurate enough to recover the true treatment effects, especially ITE. To overcome this issue, we take into account additional data observations from all the sources $\mathrm{s} \in \mathcal{S}$. Consequently, we learn the distributions in Eq. (8) using training data of the target population and all the source populations. In the subsequent sections, we present how to *adaptively* approximate these distributions.

Identification. The causal effects are unidentifiable when the confounders are unobserved. These, however, are identifiable if we can learn the confounders from the proxy variable. Louizos et al. [24] (Theorem 1) suggested that if we can learn the joint distribution $p(\mathbf{z}^{\mathrm{t}}, \mathbf{x}^{\mathrm{t}}, y^{\mathrm{t}}, w^{\mathrm{t}})$, then the causal effects are identifiable. In this case, one can use variational auto-encoder (VAE) to recover the latent confounders. This is because a VAE can learn a rich class of latent-variable models [24, 26]. Identification of our work closely follows from Louizos et al. [24]. We developed an *adaptive* variational inference algorithm that utilizes multiple data sources to learn the distributions in Eq. (8), and thus the joint distribution $p(\mathbf{z}^{\mathrm{t}}, \mathbf{x}^{\mathrm{t}}, y^{\mathrm{t}}, w^{\mathrm{t}})$ is learned. This results in identifiability of the proposed model.

3.4 Learning the Proposed Model

In this section, we present our AdaTRANS method to learn the distributions in Eq. (8) using observational data from the source populations and the target population. We propose three levels of knowledge transfer via three sets of learnable similarity coefficients to learn these distributions. Our method is different from the existing works in that it models the nonlinear functions using adaptive transfer kernel function and an augmented-representer theorem. It requires no prior knowledge on data discrepancy of the source and the target population, and fewer tuning of the model architecture.

3.4.1 1^{st} Level: Learning Distributions Involving Latent Confounders.

We start with learning $p(\mathbf{z}_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}}, w_i^{\mathrm{t}}, y_i^{\mathrm{t}})$ and $p(y_i^{\mathrm{t}}|\mathbf{z}_i^{\mathrm{t}}, w_i^{\mathrm{t}})$ that include the latent confounders. Since exact inference is intractable because of the existence of latent confounders, so we maximize evidence lower bound (ELBO) of the marginal likelihood:

$$\mathcal{L} := \sum_{\mathrm{d}\in\{\mathrm{t}\}\cup\mathcal{S}} \Big\{ E_{\mathbf{z}^{\mathrm{d}}\sim q(\mathbf{z}^{\mathrm{d}}|\cdot)} \big[\log p(\mathbf{y}^{\mathrm{d}}|\mathbf{w}^{\mathrm{d}}, \mathbf{z}^{\mathrm{d}}) + \log p(\mathbf{w}^{\mathrm{d}}|\mathbf{z}^{\mathrm{d}}) \\ \log p(\mathbf{x}^{\mathrm{d}}|\mathbf{z}^{\mathrm{d}}) \big] - D_{\mathrm{KL}} \big[q(\mathbf{z}^{\mathrm{d}}|\cdot) \| p(\mathbf{z}^{\mathrm{d}}) \big] \Big\}, \tag{9}$$

where we use bold-face notation $\mathbf{y}^{\mathrm{d}} = [y_1^{\mathrm{d}}, ..., y_{n_d}^{\mathrm{d}}]^\top$ to denote the vector of all training outcomes in population d, and similarly for the covariates $\mathbf{x}^{\mathrm{d}}$, treatments $\mathbf{w}^{\mathrm{d}}$ and latent confounders $\mathbf{z}^{\mathrm{d}}$. The ELBO $\mathcal{L}$ is computed with training data from the source populations and the target population.

The first component in $\mathcal{L}$ can be obtained from the structural equations in Eqs. (2)-(6). In particular, $p(\mathbf{y}^{\mathrm{d}}|\mathbf{w}^{\mathrm{d}}, \mathbf{z}^{\mathrm{d}})$ is from the structural equation of the outcome in Eq. (2) or (3). $p(\mathbf{w}^{\mathrm{d}}|\mathbf{z}^{\mathrm{d}})$ is from Eq. (4), and $p(\mathbf{x}^{\mathrm{d}}|\mathbf{z}^{\mathrm{d}})$ is from Eq. (5) and/or (6).

The notation $q(\mathbf{z}_i^{\mathrm{d}}|\cdot) = q(\mathbf{z}_i^{\mathrm{d}}|\mathbf{x}_i^{\mathrm{d}}, w_i^{\mathrm{d}}, y_i^{\mathrm{d}})$ denotes the variational posterior distribution. To be computationally tractable, we use mean-field approximation, i.e., $q(\mathbf{z}|\cdot) = \prod_{\mathrm{d}} \prod_i q(\mathbf{z}_i^{\mathrm{d}}|\mathbf{x}_i^{\mathrm{d}}, w_i^{\mathrm{d}}, y_i^{\mathrm{d}})$, and set the variational posterior to be a normal distribution:

$$q(\mathbf{z}|\cdot) = \prod_{\mathrm{d}\in\{\mathrm{t}\}\cup\mathcal{S}} \prod_{i=1}^{n_{\mathrm{d}}} \mathrm{N}(\mathbf{z}_i^{\mathrm{d}}; f_q(\mathbf{x}_i^{\mathrm{d}}, w_i^{\mathrm{d}}, y_i^{\mathrm{d}}), \sigma_q^2).$$

The function $f_q(\cdot)$ is as follows:

$$f_q(\mathbf{x}_i^{\mathrm{d}}, w_i^{\mathrm{d}}, y_i^{\mathrm{d}}) = w_i^{\mathrm{d}} f_{q_1}(x_i^{\mathrm{d}}, y_i^{\mathrm{d}}) + (1 - w_i^{\mathrm{d}}) f_{q_0}(x_i^{\mathrm{d}}, y_i^{\mathrm{d}}),$$

where $f_{q_0} : \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{F}_{q_0}$ and $f_{q_1} : \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{F}_{q_1}$ with $\mathcal{X}$ and $\mathcal{Y}$ are the sets containing $\mathbf{x}_i^{\mathrm{d}}$ and y_i^{d}, respectively, and $\mathcal{F}_{q_0}$, $\mathcal{F}_{q_1}$ are Hilbert spaces. From Eq. (9), optimizing $\mathcal{L}$ allows us to learn the distributions on the target population $p(y_i^{\mathrm{t}}|\mathbf{z}_i^{\mathrm{t}}, w_i^{\mathrm{t}})$ and $q(\mathbf{z}_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}}, w_i^{\mathrm{t}}, y_i^{\mathrm{t}})$ (which approximates $p(\mathbf{z}_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}}, w_i^{\mathrm{t}}, y_i^{\mathrm{t}})$). Learning these distributions leads to learning the functions f_c where $c \in \{y_0, y_1, q_0, q_1, x, w\}$ and the hyperparameters.

Adaptive transfer learning. To learn the aforementioned distributions, we first formalize an empirical risk and then optimize its regularized objective function. To proceed, we first draw L samples of latent confounders using this relation $\mathbf{z}_i^{\mathrm{d}}[l] = f_q(\mathbf{x}_i^{\mathrm{d}}, w_i^{\mathrm{d}}, y_i^{\mathrm{d}}) + \sigma_q \boldsymbol{\epsilon}_i^{\mathrm{d}}[l]$, where $\boldsymbol{\epsilon}_i^{\mathrm{d}}[l]$ denotes a vector of d_z dimensions with each element drawn from the standard normal distribution. With this procedure, we form an augmented training dataset as follows

$$\mathcal{D} = \bigcup_{\mathrm{d}\in\{\mathrm{t}\}\cup\mathcal{S}} \bigcup_{i=1}^{n_{\mathrm{d}}} \bigcup_{l=1}^{L} \Big\{ (y_i^{\mathrm{d}}, w_i^{\mathrm{d}}, \mathbf{x}_i^{\mathrm{d}}, \mathbf{z}_i^{\mathrm{d}}[l]) \Big\}.$$

The augmented training dataset $\mathcal{D}$ is the combined data from all populations $\mathrm{d} \in \{\mathrm{t}\} \cup \mathcal{S}$. This dataset is then substituted into the ELBO $\mathcal{L}$ to obtain the Monte-Carlo approximation of $\mathcal{L}$, whose negative quantity is the empirical risk.

Lemma 1. *Let $\widehat{\mathcal{L}}$ be the empirical risk obtained from the negative of the ELBO $\mathcal{L}$. Let τ_c ($c \in \{y_0, y_1, q_0, q_1, x, w\}$) be kernel functions and $\mathcal{H}_c$ be their associated reproducing kernel Hilbert spaces* (RKHSs). *Consider minimizing*

$$J = \widehat{\mathcal{L}}(f_{y_0}, f_{y_1}, f_{q_0}, f_{q_1}, f_x, f_w) + \sum\nolimits_c \gamma_c \|f_c\|_{\mathcal{H}_c}^2 \tag{10}$$

with respect to f_c ($c \in \{y_0, y_1, q_0, q_1, x, w\}$), where $\gamma_c \in \mathbb{R}^+$. The minimizer of J leads to

$$f_c(\mathbf{v}_i) = \sum\nolimits_j \tau_c(\mathbf{v}_i, \mathbf{v}_j) \boldsymbol{\alpha}_j^c, \quad c \in \{y_0, y_1, q_0, q_1, x, w\},$$

where $\mathbf{v}_{(\cdot)}$ is the input obtained from the tuples in $\mathcal{D}$. In particular, $\mathbf{v}_{(\cdot)} = \mathbf{z}_i^d[l]$ for $c = \{y_0, y_1, x, w\}$ and $\mathbf{v}_{(\cdot)} = (\mathbf{x}_i^d, y_i^d)$ for $c \in \{q_0, q_1\}$. The coefficients $\boldsymbol{\alpha}_j^c$ are vectors in the Hilbert space $\mathcal{F}_c$.

Minimizing J with respect to $\boldsymbol{\alpha}_j^c$ and parameters of specific kernel functions (e.g., Radial basis function kernel, Matérn kernel, Rational Quadratic kernel, etc.), we obtain the functions f_c and thus the distributions $p(y_i^{\mathrm{t}}|\mathbf{z}_i^{\mathrm{t}}, w_i^{\mathrm{t}})$ and $q(\mathbf{z}_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}}, w_i^{\mathrm{t}}, y_i^{\mathrm{t}})$. The proof of Lemma 1 is presented in Appendix B.

Transferable kernel function. The controlling of knowledge transfer is via the kernel functions τ_c in Lemma 1. Let d_1 and d_2 be two populations, i.e., $\mathrm{d}_1, \mathrm{d}_2 \in \{\mathrm{t}\} \cup \mathcal{S}$. Let $\mathbf{v}_i^{\mathrm{d}_1}$ and $\mathbf{v}_j^{\mathrm{d}_2}$ be two data points obtained from two tuples of the dataset $\mathcal{D}$ ($\mathbf{v}_i^{\mathrm{d}_1}$ and $\mathbf{v}_j^{\mathrm{d}_2}$ can be portions of the tuple depending on the input to the kernel function τ_c). We propose to use

$$\tau_c(\mathbf{v}_i^{\mathrm{d}_1}, \mathbf{v}_j^{\mathrm{d}_2}) = \begin{cases} \lambda^{(\mathrm{d}_1,\mathrm{d}_2)} k_c(\mathbf{v}_i^{\mathrm{d}_1}, \mathbf{v}_j^{\mathrm{d}_2}), & \text{if } \mathrm{d}_1 \neq \mathrm{d}_2, \\ k_c(\mathbf{v}_i^{\mathrm{d}_1}, \mathbf{v}_j^{\mathrm{d}_2}), & \text{otherwise,} \end{cases} \tag{11}$$

where we use a learnable parametric coefficient $\lambda^{(\mathrm{d}_1,\mathrm{d}_2)} \in [0, 1]$, which we call the *transfer factor*, to re-weight the similarity of the two populations d_1 and d_2. Since there are m source populations and one target population, we would have $m(m + 1)/2$ transfer factors. This is the first level of knowledge transfer in our method. $k_c(\cdot, \cdot)$ in Eq. (11) is a typical kernel function such as Matérn kernel, radial basis function (RBF) kernel, or rational quadratic (RQ) kernel. The transfer factors $\lambda^{(\mathrm{d}_1,\mathrm{d}_2)}$ ($\forall \mathrm{d}_1 \neq \mathrm{d}_2$) are learned together with the parameters $\boldsymbol{\alpha}^c_{(\cdot)}$ and hyperparameters of the kernel function $k_c(\cdot, \cdot)$. When $\lambda^{(\mathrm{d}_1,\mathrm{d}_2)} = 1$, it indicates that the two populations are highly related, this is equivalent to simply combining the source and target data. When $\lambda^{(\mathrm{d}_1,\mathrm{d}_2)} = 0$, the two populations are completely unrelated, it corresponds to learning the desired distributions with target data only. For $0 < \lambda^{(\mathrm{d}_1,\mathrm{d}_2)} < 1$, the desired distributions on the target population are learned with target data and partial of the source data.

Lemma 2. *Let $\boldsymbol{\alpha}_j^{q_0}$ and $\boldsymbol{\alpha}_j^{q_1}$ be fixed. Then, the objective function J in Lemma 1 is convex with respect to $\boldsymbol{\alpha}_j^{\mathrm{c}}$ for all $c \in \{y_0, y_1, x, w\}$.*

The proof of Lemma 2 is presented in Appendix C. Lemma 2 implies that if $\boldsymbol{\alpha}^{q_0}$ and $\boldsymbol{\alpha}^{q_1}$ reach their convex hull, J will reach its minimal point. This is because the non-convexity of J is induced by $\boldsymbol{\alpha}^{q_0}$ and $\boldsymbol{\alpha}^{q_1}$. This result shows that we should try different random initialization on $\boldsymbol{\alpha}^{q_0}$ and $\boldsymbol{\alpha}^{q_1}$ rather than the other parameters when optimizing J.

3.4.2 2^{nd} Level: Learning Conditional Distribution of the Outcome.

This section presents the learning of $p(y_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}}, w_i^{\mathrm{t}})$. We denote its approximation as $\tilde{p}(y_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}}, w_i^{\mathrm{t}})$. Here we also adaptively transfer knowledge from source populations to the target population. As all the variables involved are observed, we learn these distributions by maximizing log-likelihood of the observed data. Specifically, we model

$$\tilde{p}(y_i^{\mathrm{d}}|\mathbf{x}_i^{\mathrm{d}}, w_i^{\mathrm{d}}) = \mathsf{N}\big(y_i^{\mathrm{d}}; g(\mathbf{x}_i^{\mathrm{d}}, w_i^{\mathrm{d}}), \tilde{\sigma}_y^2\big)$$

for continuous outcome, and

$$\tilde{p}(y_i^{\mathrm{d}}|\mathbf{x}_i^{\mathrm{d}}, w_i^{\mathrm{d}}) = \mathsf{Bern}\big(y_i^{\mathrm{d}}; \varphi(g(\mathbf{x}_i^{\mathrm{d}}, w_i^{\mathrm{d}}))\big)$$

for binary outcome, where $\tilde{\sigma}_y^2$ is the noise variance and $\varphi(\cdot)$ is the logistic function. We model $g(\mathbf{x}_i^{\mathrm{d}}, w_i^{\mathrm{d}}) = w_i^{\mathrm{d}}\, g_1(\mathbf{x}_i^{\mathrm{d}}) + (1 - w_i^{\mathrm{d}})\, g_0(\mathbf{x}_i^{\mathrm{d}})$, where $g_0 \colon \mathcal{X} \mapsto \mathcal{F}_{y_0}$ and $g_1 \colon \mathcal{X} \mapsto \mathcal{F}_{y_1}$ are functions modelling the outcome when the treatment $w_i^{\mathrm{d}} = 0$ and $w_i^{\mathrm{d}} = 1$, respectively. We obtain the regularized empirical risk as follows:

$$J_y = \widehat{\mathcal{L}}_y(g_0, g_1) + \gamma_{y_0} \|g_0\|^2_{\mathcal{V}_y} + \gamma_{y_1} \|g_1\|^2_{\mathcal{V}_y}, \tag{12}$$

where $\widehat{\mathcal{L}}_y(\cdot)$ is the negative log-likelihood, $\mathcal{V}_y$ is a reproducing kernel Hilbert space associated a kernel function $\psi_y(\mathbf{x}_i^{\mathrm{d}_1}, \mathbf{x}_j^{\mathrm{d}_2})$, and $\gamma_{y_0}, \gamma_{y_1} \in \mathbb{R}^+$. Herein, $\mathrm{d}_1, \mathrm{d}_2 \in \{\mathrm{t}\} \cup \mathcal{S}$ are two populations. So we use another set of transfer factors $\delta^{(\mathrm{d}_1,\mathrm{d}_2)} \in [0, 1]$ to re-weight the cross-population similarity. The transfer factors here are learned from data.

Algorithm 1: Learning the model

```
Input:  $\{(y_i^{\mathrm{t}}, w_i^{\mathrm{t}}, \mathbf{x}_i^{\mathrm{t}})\}_{i=1}^{n_{\mathrm{t}}}$ and $\{(y_i^{\mathrm{s}}, w_i^{\mathrm{s}}, \mathbf{x}_i^{\mathrm{s}})\}_{i=1}^{n_{\mathrm{s}}}$ for all s $\in \mathcal{S}$.
begin
    Optimize $J$ in Eq. (10) to obtain $p(y_i^{\mathrm{t}}|\mathbf{z}_i^{\mathrm{t}}, w_i^{\mathrm{t}})$ & $q(\mathbf{z}_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}}, w_i^{\mathrm{t}}, y_i^{\mathrm{t}})$;
    Optimize $J_y$ in Eq. (12) to obtain $\tilde{p}(y_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}}, w_i^{\mathrm{t}})$;
    Optimize $J_w$ (Section 3.4.3) to obtain $\tilde{p}(w_i^{\mathrm{t}})|\mathbf{x}_i^{\mathrm{t}})$;
```

Algorithm 2: Estimating causal effects

```
Input:  Set of covariates $\{\mathbf{x}_{*i}^{\mathrm{t}}\}_{i=1}^{n_*}$ of $n_*$ individuals.
        The distributions obtained from Algorithm 1.
begin
    $L \leftarrow \emptyset$;
    for $i \leftarrow 1$ to $n_*$ do
        $L_0 \leftarrow \emptyset$ and $L_1 \leftarrow \emptyset$;
        for $j \leftarrow 1$ to $M$ do
            Draw a sample $w$ from $\tilde{p}(w_i^{\mathrm{t}})|\mathbf{x}_i^{\mathrm{t}} = \mathbf{x}_{*i}^{\mathrm{t}})$;
            Draw a sample $y$ from $\tilde{p}(y_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}} = \mathbf{x}_{*i}^{\mathrm{t}}, w_i^{\mathrm{t}} = w)$;
            Draw $z$ from $q(\mathbf{z}_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}} = \mathbf{x}_{*i}^{\mathrm{t}}, w_i^{\mathrm{t}} = w, y_i^{\mathrm{t}} = y)$;
            Draw $y_0$ from $p(y_i^{\mathrm{t}}|\mathbf{z}_i^{\mathrm{t}} = z, w_i^{\mathrm{t}} = 0)$ and add it to $L_0$;
            Draw $y_1$ from $p(y_i^{\mathrm{t}}|\mathbf{z}_i^{\mathrm{t}} = z, w_i^{\mathrm{t}} = 1)$ and add it to $L_1$;
        Compute ITE:   $\widehat{\mathrm{ite}}_i \leftarrow (\sum_{y_1 \in L_1} y_1 - \sum_{y_0 \in L_0} y_0)/M$;
        Add $\widehat{\mathrm{ite}}_i$ to $L$;
    Compute ATE:   $\widehat{\mathrm{ate}} = \frac{1}{n_*} \sum_{i=1}^{n_*} \widehat{\mathrm{ite}}_i$;
    return $\widehat{\mathrm{ate}}$, $\{\widehat{\mathrm{ite}}_i\}_{i=1}^{n_*}$
```

3.4.3 3^{rd} Level: Learning Conditional Distribution of the Treatment.

We denote the approximation of $p(w_i^{\mathrm{t}}|\mathbf{x}_i^{\mathrm{t}})$ as $\tilde{p}(w_i^{\mathrm{d}}|\mathbf{x}_i^{\mathrm{d}})$. Since the treatment is binary, it can be modeled by the Bernoulli distribution. Concretely, $\tilde{p}(w_i^{\mathrm{d}}|\mathbf{x}_i^{\mathrm{d}}) = \mathsf{Bern}\big(w_i^{\mathrm{d}}; \varphi(h(\mathbf{x}_i^{\mathrm{d}}))\big)$, where $h \colon \mathcal{X} \mapsto \mathcal{F}_w$. Similar to the above, the regularized empirical risk obtained from the negative log-likelihood is $J_w = \widehat{\mathcal{L}}_w(h) + \gamma_w \|h\|^2_{\mathcal{V}_w}$, where $\mathcal{V}_w$ is a reproducing kernel Hilbert space associated with a kernel function $\psi_w(\mathbf{x}_i^{\mathrm{d}_1}, \mathbf{x}_j^{\mathrm{d}_2})$. Here we use another set of transfer factors $\eta^{(\mathrm{d}_1,\mathrm{d}_2)} \in [0, 1]$.

To sum up, we have introduced three levels of knowledge transfer from multiple source populations to the target population via an augmented-representer theorem algorithm and transfer kernels. This enables an adaptive causal inference algorithm to estimate causal effects in the target population.

We summarize the training steps in Algorithm 1 and the steps to estimate causal effects in Algorithm 2.

4 EXPERIMENTS

Baselines and the aims of our experiments. In this section, we first perform a set of experiments to verify the effectiveness of our proposed model (AdaTRANS) in adaptively transferring knowledge from source populations to the target population, and thus improving the estimation of the treatment effects of interest. Here we aim to illustrate the importance of our proposed adaptive transfer learning method in estimating causal effects. Our second analysis is to compare the proposed method against some recent baselines including BART [17], CFRNet [47], CEVAE [24], OrthoRF [31], SITE [53], X-learner [21], and R-learner [30]. These baselines do not consider data scarcity problem in the target population. The aim of this analysis is to show the efficacy of our method when some sources of data are available.

The setups of neural networks in Louizos et al. [24] (CEVAE) and [53] (SITE) closely follow that of Shalit et al. [47] (CFRNet). Thus we also use these settings in our experiments. In particular, we use fully connected networks with activation function `ELU` and use the same number of hidden nodes in each hidden layer. We fine-tune all the networks with $\{1, 2, \ldots, 6\}$ hidden layers, $\{50, 100, 200\}$ number of nodes per layer, and learning rate in $\{1e\text{-}1, 1e\text{-}2, 1e\text{-}3, 1e\text{-}4\}$. We reuse the code of these methods which are available online. For implementation of BART [17], we use package `BartPy` which is also available online. For X-learner [21] and R-learner [30], we use package `causalml` [11]. In both methods, we use `xgboost.XGBClassifier` as learners for binary outcomes and `xgboost.XGBRegressor` as learners for continuous outcomes. For OrthoRF [31], we use package `econml` [28].

Evaluation metrics. Two comparison metrics are used in our experiments: precision in estimation of heterogeneous effects (PEHE) [17]: $\epsilon_{\text{PEHE}} = E[((y_1 - y_0) - (\hat{y}_1 - \hat{y}_0))^2]$ for evaluating ITE, and absolute error: $\epsilon_{\text{ATE}} = |E[y_1 - y_0] - E[\hat{y}_1 - \hat{y}_0]|$ for evaluating ATE, where y_0, y_1 are the ground truth of outcomes from the intervention and $\hat{y}_0, \hat{y}_1$ are their estimates. The reported numbers are the out-of-sample mean and standard error over 10 replicates of the data with different random initializations of the training algorithm.

4.1 Synthetic Data

Data description. Obtaining the ground-truth of causal inference problems is challenging, and thus most of recent methods utilize synthetic or semi-synthetic datasets for evaluation. In this experiment, we generate synthetic datasets, each comprises of data from m source populations ($\boldsymbol{\mathcal{S}} = \{\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_m\}$) and one target (t) population. Our aim is to show that the estimated ITE and ATE on the target population is closer to the true values when utilizing knowledge transferred from the source population. For each individual i in the population $\mathrm{d} \in \{\mathrm{t}\} \cup \boldsymbol{\mathcal{S}}$, we draw the latent confounder $\mathbf{z}_i^{\mathrm{d}}$, the proxy variable $\mathbf{x}_i^{\mathrm{d}}$, the treatment w_i^{d} and the outcome y_i^{d} using the following equations

$$\mathbf{z}_i^{\mathrm{d}} \sim \mathsf{N}(\mathbf{0}, \sigma_z^2 \mathbf{I}_2), \qquad x_{ij}^{\mathrm{d}} \sim \mathsf{Bern}(\varphi(a_{0j} + (\mathbf{z}_i^{\mathrm{d}})^\top \boldsymbol{a}_{1j})),$$
$$w_i^{\mathrm{d}} \sim \mathsf{Bern}(\varphi(b_0 + (\mathbf{z}_i^{\mathrm{d}})^\top \boldsymbol{b}_1^{\mathrm{d}})), \qquad y_i^{\mathrm{d}}(0) \sim \mathsf{N}(\zeta(c_0 + (\mathbf{z}_i^{\mathrm{d}})^\top \boldsymbol{c}_1^{\mathrm{d}}), \sigma_y^2),$$
$$y_i^{\mathrm{d}}(1) \sim \mathsf{N}(\zeta(d_0 + (\mathbf{z}_i^{\mathrm{d}})^\top \boldsymbol{d}_1^{\mathrm{d}}), \sigma_y^2),$$

where $\varphi(\cdot)$ is the standard logistic function, and $\zeta(\cdot)$ is the softplus function. We randomly set the ground truth parameters $(\sigma_z, \sigma_y) = (\sqrt{8}, \sqrt{2})$, $(b_0, c_0, d_0) = (0.5, 0.7, 2.0)$, and draw the ground truth $a_{0j} \sim \mathsf{N}(0, 2)$ and $\boldsymbol{a}_{1j} \sim \mathsf{N}(0, 2 \cdot \mathbf{I}_2)$ (for $j = 1, 2, \ldots, 30$). Herein, the number of dimensions of the latent confounder $\mathbf{z}_i^{\mathrm{d}}$ is $d_z = 2$ and the number of proxy variables is $d_x = 30$. The parameters $\boldsymbol{b}_1^{\mathrm{d}}, \boldsymbol{c}_1^{\mathrm{d}}, \boldsymbol{d}_1^{\mathrm{d}}$ on different population d would have different ground truth values. The aim is to simulate the difference of the data distribution in different populations, which showcases the effectiveness of our model. We will describe these three parameters in the specific analyses. From the above simulation, we obtain $y_i^{\mathrm{d}} = w_i^{\mathrm{d}}\, y_i^{\mathrm{d}}(1) + (1 - w_i^{\mathrm{d}})\, y_i^{\mathrm{d}}(0)$, i.e., $y_i^{\mathrm{d}} = y_i^{\mathrm{d}}(1)$ when $w_i^{\mathrm{d}} = 1$ and $y_i^{\mathrm{d}} = y_i^{\mathrm{d}}(0)$ when $w_i^{\mathrm{d}} = 0$. For each individual i, we only keep $(y_i^{\mathrm{d}}, w_i^{\mathrm{d}}, \mathbf{x}_i^{\mathrm{d}})$ as the observed data, and discard $\mathbf{z}_i^{\mathrm{d}}$. For each population d (source or target), we simulate a set of $n_{\mathrm{d}} = 1000$ individuals. Thus, the total number source observations is $m \times 1000$. For the target data, since this data is scarce, we only use 50 for training, 100 for validation and 850 for testing. In the subsequent sections, we present the performance analysis of AdaTRANS (the proposed method) compared with the baselines on this dataset.

4.1.1 *Analysis I: The Importance of Adaptive Transfer.*

Additional setups on the synthetic data. To verify the proposed adaptively causal transfer learning model, here we use one source population s ($m = 1$) and one target population t. We will analyse on multi-source ($m > 1$) in the subsequent sections. In this experiment, we have two sets of ground truth parameters ($\boldsymbol{b}_1^{\mathrm{s}}$, $\boldsymbol{c}_1^{\mathrm{s}}$, $\boldsymbol{d}_1^{\mathrm{s}}$) and ($\boldsymbol{b}_1^{\mathrm{t}}$, $\boldsymbol{c}_1^{\mathrm{t}}$, $\boldsymbol{d}_1^{\mathrm{t}}$) and we set them differently as follows

$$\boldsymbol{b}_1^{\mathrm{t}} = [1.1, 1.7]^\top, \qquad \boldsymbol{c}_1^{\mathrm{t}} = [1.5, 1.8]^\top, \qquad \boldsymbol{d}_1^{\mathrm{t}} = [1.5, 2.8]^\top$$
$$\boldsymbol{b}_1^{\mathrm{s}} = \boldsymbol{b}_1^{\mathrm{t}} + \Delta^{\mathrm{s}}[1, 1]^\top, \quad \boldsymbol{c}_1^{\mathrm{s}} = \boldsymbol{c}_1^{\mathrm{t}} + \Delta^{\mathrm{s}}[1, 1]^\top, \quad \boldsymbol{d}_1^{\mathrm{s}} = \boldsymbol{d}_1^{\mathrm{t}} + \Delta^{\mathrm{s}}[1, 1]^\top,$$

where we vary $\Delta^{\mathrm{s}} \in \{0.0, 0.5, 1.0, 1.5, 2.0\}$ to obtain different instances of the source data, which simulates the similarity/dissimilarity between the source and the target. Here, Δ^{s} denotes the discrepancy between the source and target population.

Results and discussion. Figure 2 presents the performance of AdaTRANS (the proposed method) compared with the case of full transfer (transfer factor is set to 1) and the case of no transfer (transfer factor is set to 0). The figure clearly shows that our proposed method can adaptively learn the transfer factors $\lambda^{(\mathrm{t,s})}$, $\delta^{(\mathrm{t,s})}$, $\eta^{(\mathrm{t,s})}$ to control the knowledge transfer from source to target data. In general, the more discrepant the source and target population is, the lower transfer factors are. Thus, it results in better performance of our adaptive transfer than full transfer and no transfer. On our second analysis, we study the importance of each level of knowledge transfer in our proposed method. So we turn off one of the transferring level and observe the performance. Figure 3 illustrates the performance of each case compared with 'adaptive transfer' on all levels. The figure shows that the first level (learning $\lambda^{(\mathrm{t,s})}$) is the most important as the performance would reduce more when we turn off this parameter (set $\lambda^{(\mathrm{t,s})} = 0$). This is the transferring level of learning distributions regarding latent confounders. Hence, learning latent confounders plays an important role in estimating causal effects.

4.1.2 *Analysis II: Multiple Sources.*

Additional setups on the synthetic data. In this experiment, we simulate $m = 4$ data sources where the ground truth of $\boldsymbol{b}_1^{\mathrm{t}}, \boldsymbol{c}_1^{\mathrm{t}}, \boldsymbol{d}_1^{\mathrm{t}}$ are set as in the previous experiment in Analysis I. The other parameters $\boldsymbol{b}_1^{\mathrm{s}}, \boldsymbol{c}_1^{\mathrm{s}}, \boldsymbol{d}_1^{\mathrm{s}}$ (where $\mathrm{s} \in \{\mathrm{s}_1, \mathrm{s}_2, \mathrm{s}_3, \mathrm{s}_4\}$) are set as $(\Delta^{\mathrm{s}_1}, \Delta^{\mathrm{s}_2}, \Delta^{\mathrm{s}_3}, \Delta^{\mathrm{s}_4}) = (2.0, 1.5, 1.0, 0.5)$. Hence, different source populations have different levels of discrepancy to the target one.

Results and discussion. Figure 4 reports the performance of AdaTRANS (adaptive transfer) compared with full transfer and no transfer. In Figure 4, #1 source means that the source s_1 is used, #2 source means that $\mathrm{s}_1, \mathrm{s}_2$ are used, and so on, i.e., more sources that are

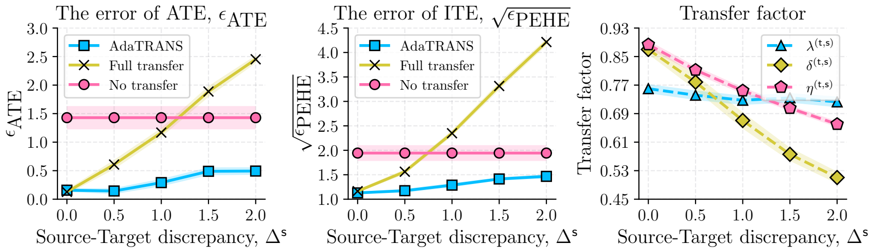


Figure 2: Adaptively causal transfer learning analysis.

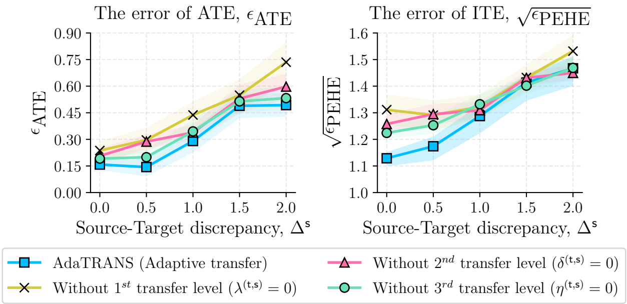


Figure 3: Partially causal transfer analysis.

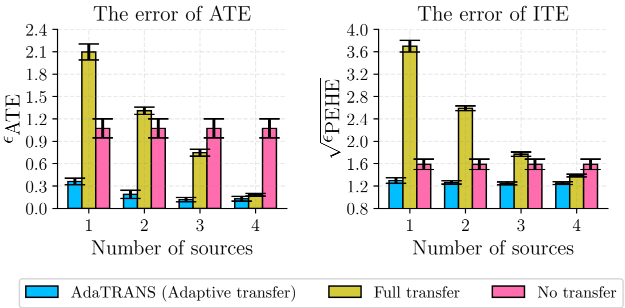


Figure 4: Multi-source causal transfer analysis

more and more similar to the target are added. The figure shows that the more data sources, the better performance of our model. In the case of full transfer, the errors also reduce when more sources are added. This is because we are adding more sources that are more and more similar to the target data.

4.1.3 *Analysis III: Compare with the Baselines.*

Two setups of the baselines. In this section, we compare AdaTRANS with the baselines. For each baseline, we train with two cases as follows. (1) In the first case, we combine sources and target data by adding a categorical covariate to indicate the population of each entry in the dataset. This categorical covariate is then transformed into a '1-hot vector' for training the models. (2) In the second case, we combine source and target data by 'stacking' them, i.e., there is no additional covariate. The data we use in this analysis is the one simulated in Analysis II.

Table 1: Out-of-sample error on synthetic dataset with different number of data sources. The dashes (—) in '1-hot' indicate that the numbers are the same as those of 'stack'.

Method	The error of ITE ($\sqrt{\epsilon_{PEHE}}$)			The error of ATE (ϵ_{ATE})		
	0-source	2-sources	4-sources	0-source	2-sources	4-sources
CEVAE$_{stack}$	3.1±.30	4.6±.39	4.8±.40	1.7±.29	2.8±.30	2.5±.26
CFRNet$_{stack}$	4.6±.51	8.9±.50	6.0±.19	1.6±.41	6.1±.48	4.0±.17
SITE$_{stack}$	6.0±.98	8.9±.61	7.5±.60	3.3±.67	6.4±.79	5.0±.76
BART$_{stack}$	2.5±.06	2.3±.03	2.2±.06	1.2±.13	0.7±.08	0.6±.09
R-learner$_{stack}$	3.0±.27	2.2±.11	1.8±.09	1.4±.35	1.2±.17	1.0±.10
X-learner$_{stack}$	**2.0±.13**	2.2±.12	1.9±.13	**1.0±.17**	1.0±.11	1.1±.13
OrthoRF$_{stack}$	6.2±.40	2.4±.03	2.2±.03	1.2±.37	**0.5±.08**	0.6±.06
CEVAE$_{1\text{-hot}}$	—	5.0±.43	3.3±.12	—	3.1±.42	1.9±.23
CFRNet$_{1\text{-hot}}$	—	4.4±.26	3.3±.21	—	3.3±.26	2.1±.17
SITE$_{1\text{-hot}}$	—	5.8±.99	3.2±.25	—	3.4±.67	2.1±.21
BART$_{1\text{-hot}}$	—	2.3±.03	2.2±.04	—	0.7±.10	**0.4±.10**
R-learner$_{1\text{-hot}}$	—	2.0±.07	**1.7±.15**	—	0.8±.15	0.8±.20
X-learner$_{1\text{-hot}}$	—	**1.9±.12**	1.8±.10	—	0.7±.13	0.6±.12
OrthoRF$_{1\text{-hot}}$	—	5.5±.30	4.1±.16	—	3.9±.22	2.6±.17
AdaTRANS	**1.6±.09**	**1.3±.03**	**1.3±.02**	**1.1±.13**	**0.2±.05**	**0.1±.03**

Results and discussion. Table 1 reports the performance of each method in estimating ATE and ITE. In term of predicting ATE, the figures (three last columns) show significant improvement when adding more data sources (for 2 and 4 sources). In term of predicting ITE, the figures also show that AdaTRANS achieves lower error even with 0-source. This might be because of the advantage of using kernel method, which give a flexible modelling for non-linear functions. The figures also reveal that negative transfer happened in CEVAE, CRFNet and SITE since the performance of these method decrease when adding more data sources whose distributions are different from that of the target. We also observe that positive transfer appears in BART, R-learner, X-learner, and OrthoRF. In addition, the baselines trained with combination of sources and target data by adding a categorical covariate to indicate the population (1h) tends to give better performance than the baselines of stacking datasets (st). This is because the 1h case would have access to the source, hence reducing negative transfer.

4.2 Twins Dataset

Data description. The Twins dataset contains multiple records of twin births in the US from 1989 to 1991 [24]. An abstract treatment $w_i = 1$ corresponds to the twin born with heavier weight and likewise, $w_i = 0$ corresponds to the twin born with lighter weight. The outcome corresponds to the mortality of each of the twins in their first year of life. Since there are records for both twins, the mortality of twins has two possible outcomes (e.g., dead or alive) with respect to the treatment $w_i \in \{0, 1\}$. Following Louizos et al. [24], we focused on twins with both weighing less than 2kg.

The observational study is simulated as follows. For each pair of twins, observation regarding one of them is randomly excluded. The entire dataset is then partitioned into two sets: source and target data. The source data accounts for 81% (3921 entries) and the target data account for 19% (900 entries). In the target data, we

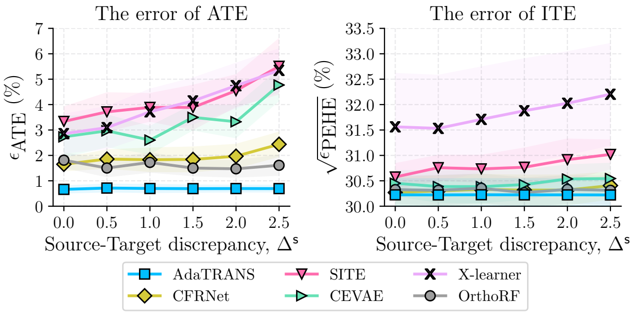


Figure 5: Out-of-sample error of ATE and ITE on Twins dataset.

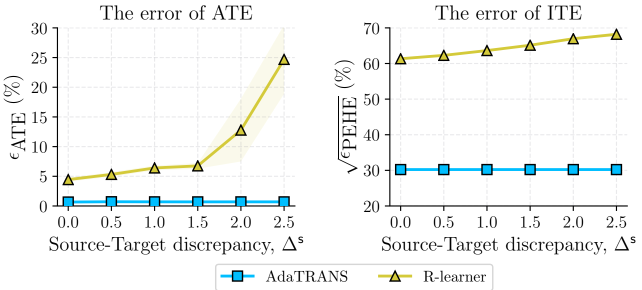


Figure 6: Out-of-sample error of ATE and ITE on Twins dataset for R-learner vs. AdaTRANS.

use 9-fold cross-validation with 100 entries for training, 100 for validation, and 700 for testing.

Simulation of latent confounders. To simulate the case of latent confounders with proxy variables, the treatment assignment on twins is based on feature GESTAT10, which records the number of gestation weeks prior to birth and is highly correlated with the mortality outcome. We obtain the observed treatments by drawing from the following distribution $w_i^{\text{d}} \mid z_i^{\text{d}} \sim \text{Bern}(\varphi(b^{\text{d}}(0.1z_i^{\text{d}} - 0.1)))$, where $\text{d} \in \{\text{s}, \text{t}\}$ and z_i^{d} is GESTAT10. We set $b^{\text{t}} = 0.2$ and $b^{\text{s}} = b^{\text{t}} + \Delta^{\text{s}}$ to simulate discrepancy between the source and target data. We vary $\Delta^{\text{s}} \in \{0.0, 0.5, 1.0, 1.5, 2.0, 2.5\}$. Following Louizos et al. [24], we created proxies $\mathbf{x}_i^{\text{d}}$ for the hidden confounder $\mathbf{z}_i^{\text{d}}$ as follows: The 10 categories of feature GESTAT10 are encoded with one-hot encoding, and are replicated three times. We use three replications to ensure that the confounder can be recovered [follow from 3, 4, 20, 24]. The resulting target data has 30 dimensions for proxy variable $\mathbf{x}_i^{\text{d}}$.

Results and discussion. The performance of AdaTRANS and the baselines are presented in Figure 5. It can be observed that AdaTRANS achieves lower error at all levels of discrepancy between the source and the target data. The errors of SITE and X-learner in predicting ATE (ϵ_{ATE}) is high, this might be because those methods only consider observed confounders (through the unconfoundedness assumption) and ignore latent confounders. Although CEVAE can handle latent confouders, it is based on deep neural networks, and hence needs a huge dataset to achieve a good performance. SITE and CEVAE are based on neural networks, their performance depends on the tuning of hyperparameters and the optimization algorithm. Hence, they might lead to a local optima. Our method, on the other hand, models complex non-linear function using kernel functions while obtaining convex or conditionally convex objective functions (as stated in Lemma 2). Note that BART does not support binary outcomes so we do not report it on this dataset. R-learner seems to perform the worst with all of our fine-tuning setups. For a clear illustration, we report experimental results of R-learner in Figure 6.

Table 2: Out-of-sample error on IHDP dataset with different number of data sources. The dashes (—) in '1-hot' indicate that the numbers are the same as those of 'stack'.

Method	The error of ITE ($\sqrt{\epsilon_{\text{PEHE}}}$)		The error of ATE (ϵ_{ATE})	
	0-source	1-sources	0-source	1-sources
CEVAE$_{\text{stack}}$	4.38±2.11	4.09±2.01	**2.39±1.06**	1.68±0.79
CFRNet$_{\text{stack}}$	5.62±2.60	5.54±2.66	4.15±1.77	4.06±1.84
SITE$_{\text{stack}}$	5.84±2.76	5.90±2.74	4.45±1.98	4.57±1.96
BART$_{\text{stack}}$	5.44±2.68	4.37±2.29	3.85±1.88	2.25±1.26
R-learner$_{\text{stack}}$	5.47±2.49	2.93±1.12	3.05±1.88	**0.70±0.42**
X-learner$_{\text{stack}}$	**3.90±2.06**	2.64±1.09	2.48±1.61	1.00±0.49
OrthoRF$_{\text{stack}}$	4.91±2.38	2.97±1.65	3.10±1.76	2.01±1.22
CEVAE$_{\text{1-hot}}$	—	4.16±2.07	—	1.91±0.88
CFRNet$_{\text{1-hot}}$	—	5.54±2.66	—	4.05±1.84
SITE$_{\text{1-hot}}$	—	5.97±2.70	—	4.65±1.90
BART$_{\text{1-hot}}$	—	4.46±2.34	—	2.37±1.34
R-learner$_{\text{1-hot}}$	—	**2.52±0.12**	—	**0.95±0.25**
X-learner$_{\text{1-hot}}$	—	2.64±1.09	—	1.08±0.49
OrthoRF$_{\text{1-hot}}$	—	3.74±2.24	—	2.52±1.86
AdaTRANS	**3.60±2.04**	**2.46±1.09**	**1.94±1.23**	**0.70±0.20**

4.3 IHDP Dataset

Data description. IHDP (Infant Health and Development Program) dataset contains data of a study on the effect of specialist visits on children's cognitive development. This dataset has 747 data points and 25 covariates. The covariates are properties of the children and their mother's. There are two groups of children in this dataset: with and without specialist visit. We use NPCI package [12] to simulate the two outcomes of each child: one outcome for this child with specialist visit, and another outcome for the same child, but without specialist visit. Hence, we can obtain the *true* individual treatment effect. We use k-means on the covariates to divide the data into two sets. Each set is then considered as data from a population. We choose the first set as data of the target population and the other set is data of the source population.

Results and discussion. The baselines in this experiment are also trained in two cases: using '1-hot vector' in combining data and stacking the data. Table 2 shows that our method outperforms the baselines in both of the evaluation metrics. We also observe that the errors when using one source are lower than those of without using any source . In addition, the performance of the baselines when using 'one-hot vector' are worse than those of 'stack'. These results might be because the source and the target data come from the same distribution, hence adding a one-hot vector to indicate the population (source or target) for each data point would reduce effect of the source on the target population. When training with target data only (0-source), the proposed method still outperforms the baselines, this result shows the advantage of the proposed kernel-based method.

5 CONCLUSION

We have presented a knowledge transfer method to estimate ITE and ATE in a target population. Our method utilises multiple data sources and adaptively transfers knowledge from them to the target population to infer the causal effects. Our method is different from the existing works in that it models the nonlinear functions using adaptive transfer kernels and representer theorem. It requires no prior knowledge on data discrepancy of the source and the target population, and fewer tuning of the model architecture.

A potential limitation of our approach lies in the assumption that the source populations and target population share the same causal graphs and the causal effects in all populations are identifiable with their own observed data. An interesting topic for future research is to generalize the problem to the setting where causal effects in the target population may be unidentifiable. Another direction is to study the statistical guarantees of identification of the latent confounders with proxy variables.

ACKNOWLEDGMENTS

This research/project is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-016).

A EXAMPLE ON NAIVELY COMBINING DATA

We give a simple illustration on why there are biases in the causal estimands if the outcome distribution of the source and target population are different. Suppose we have one source population whose distributions of the outcomes are: $y^{\mathrm{s}}(0) \sim \mathrm{N}(100+2x, 1)$ and $y^{\mathrm{s}}(1) \sim \mathrm{N}(105+2x, 1)$. The distributions on the target population are: $y^{\mathrm{t}}(0) \sim \mathrm{N}(110+2x, 1)$ and $y^{\mathrm{t}}(1) \sim \mathrm{N}(120+2x, 1)$. Herein, we assume that x is an observed confounder. The true ATE and ITE in the target population are 10. In real-life, we cannot observe both of the outcomes, but for illustration purpose, we suppose that they are both known. A combination of the two populations would result in mixture distributions of the outcomes: $y(0) \sim (1-\pi)\mathrm{N}(100+2x, 1) + \pi\mathrm{N}(110+2x, 1)$ and $y(1) \sim (1-\pi)\mathrm{N}(105+2x, 1) + \pi\mathrm{N}(120+2x, 1)$, where $\pi \in (0, 1)$ is small and close to 0 since the target observational data is scarce and much less than that of the source population. In this case, the *ideal* estimated ATE and ITE are $E[Y(1) - Y(0)] = E[Y(1) - Y(0)|X = x] = 5(1-\pi) + 10\pi = 5 + 5\pi$. Since π is small, this estimand is much less than the true treatment effects in the target population (which is 10). In real-life, only one of the two outcomes is observed and there might be latent confounders, the causal estimand would be even worse than the above ideal estimation.

B PROOF OF LEMMA 1

PROOF. We restate that $f_{y_0} : \mathcal{Z} \mapsto \mathcal{F}_{y_0}$, $f_{y_1} : \mathcal{Z} \mapsto \mathcal{F}_{y_1}$, $f_{q_0} : \mathcal{Y} \times \mathcal{X} \to \mathcal{F}_z$, $f_{q_1} : \mathcal{Y} \times \mathcal{X} \to \mathcal{F}_z$, $f_w : \mathcal{Z} \mapsto \mathcal{F}_w$ and $f_x : \mathcal{Z} \mapsto \mathcal{F}_x$. In this work, $\mathcal{F}_{y_0} = \mathbb{R}$, $\mathcal{F}_{y_1} = \mathbb{R}$, $\mathcal{F}_w = \mathbb{R}$, $\mathcal{F}_x = \mathbb{R}^{d_x}$ and $\mathcal{F}_z = \mathbb{R}^{d_z}$. We further define $f_x = [f_{x,1},..., f_{x,d_x}]$ with $f_{x,d} : \mathcal{Z} \mapsto \mathbb{R}$ ($d = 1, \ldots, d_x$). Similarly, $f_{q_0} = [f_{q_0,1},..., f_{q_0,d_z}]$ with $f_{q_0,d} : \mathcal{Y} \times \mathcal{X} \mapsto \mathbb{R}$ ($d = 1, \ldots, d_z$) and $f_{q_1} = [f_{q_1,1},..., f_{q_1,d_z}]$ with $f_{q_1,d} : \mathcal{Y} \times \mathcal{X} \mapsto \mathbb{R}$ ($d = 1, \ldots, d_z$). Consider the subspaces $\mathcal{U}_c \subset \mathcal{H}_c$ (where $c \in \{y_0, y_1, q_0, q_q, x, w\}$):

$$
\begin{aligned}
\mathcal{U}_{y_0} &= \mathsf{span}\big\{\kappa_{y_0}(\cdot, \mathbf{z}_i^{\mathrm{d}}[l]) \;:\; \mathrm{d} \in \mathcal{S}; i=1,..., n_{\mathrm{d}}; l=1,..., L\big\},\\
\mathcal{U}_{y_1} &= \mathsf{span}\big\{\kappa_{y_1}(\cdot, \mathbf{z}_i^{\mathrm{d}}[l]) \;:\; \mathrm{d} \in \mathcal{S}; i=1,..., n_{\mathrm{d}}; l=1,..., L\big\},\\
\mathcal{U}_{x} &= \mathsf{span}\big\{\kappa_{x}(\cdot, \mathbf{z}_i^{\mathrm{d}}[l]) \;:\; \mathrm{d} \in \mathcal{S}; i=1,..., n_{\mathrm{d}}; l=1,..., L\big\},\\
\mathcal{U}_{w} &= \mathsf{span}\big\{\kappa_{w}(\cdot, \mathbf{z}_i^{\mathrm{d}}[l]) \;:\; \mathrm{d} \in \mathcal{S}; i=1,..., n_{\mathrm{d}}; l=1,..., L\big\},\\
\mathcal{U}_{q_0} &= \mathsf{span}\big\{\kappa_{q_0}(\cdot, [y_i^{\mathrm{d}}, \mathbf{x}_i^{\mathrm{d}}]) \;:\; \mathrm{d} \in \mathcal{S}; i=1,..., n_{\mathrm{d}}\big\},\\
\mathcal{U}_{q_1} &= \mathsf{span}\big\{\kappa_{q_1}(\cdot, [y_i^{\mathrm{d}}, \mathbf{x}_i^{\mathrm{d}}]) \;:\; \mathrm{d} \in \mathcal{S}; i=1,..., n_{\mathrm{d}}\big\}.
\end{aligned}
$$

We project f_{y_0}, f_{y_1}, f_w, $f_{x,d}$ ($d = 1,..., d_x$), $f_{q_0,d}$ ($d = 1,..., d_z$) and $f_{q_1,d}$ ($d = 1,..., d_z$) onto the subspaces $\mathcal{U}_{y_0}$, $\mathcal{U}_{y_1}$, $\mathcal{U}_w$, $\mathcal{U}_x$, $\mathcal{U}_{q_0}$ and $\mathcal{U}_{q_1}$, respectively, to obtain $f^{\mathrm{s}}_{y_0}$, $f^{\mathrm{s}}_{y_1}$, f^{s}_{w}, $f^{\mathrm{s}}_{x,d}$, $f^{\mathrm{s}}_{q_0,d}$ and $f^{\mathrm{s}}_{q_1,d}$, and also project them onto the perpendicular spaces of the subspaces to obtain $f^{\perp}_{y_0}$, $f^{\perp}_{y_1}$, $f^{\perp}_{w}$, $f^{\perp}_{x,d}$, $f^{\perp}_{q_0,d}$ and $f^{\perp}_{q_1,d}$. Note that $f^{\mathrm{s}}_{(\cdot)} + f^{\perp}_{(\cdot)} = f_{(\cdot)}$. Thus, $\|f_{(\cdot)}\|^2_{\mathcal{H}_{(\cdot)}} = \|f^{\mathrm{s}}_{(\cdot)}\|^2_{\mathcal{H}_{(\cdot)}} + \|f^{\perp}_{(\cdot)}\|^2_{\mathcal{H}_{(\cdot)}} \geq \|f^{\mathrm{s}}_{(\cdot)}\|^2_{\mathcal{H}_{(\cdot)}}$, which implies that $\gamma_{(\cdot)}\|f_{(\cdot)}\|^2_{\mathcal{H}_{(\cdot)}}$ is minimized if $f_{(\cdot)}$ is in its subspace $\mathcal{U}_{(\cdot)}$. (1)

Moreover, from the reproducing property, we have that

$$
\begin{aligned}
f_{y_0}(\mathbf{z}_i^{\mathrm{d}}[l]) &= \big\langle f_{y_0}, \kappa_{y_0}(\cdot, \mathbf{z}_i^{\mathrm{d}}[l])\big\rangle_{\mathcal{H}_y}\\
&= \big\langle f^{\mathrm{s}}_{y_0}, \kappa_{y_0}(\cdot, \mathbf{z}_i^{\mathrm{d}}[l])\big\rangle_{\mathcal{H}_y} + \big\langle f^{\perp}_{y_0}, \kappa_{y_0}(\cdot, \mathbf{z}_i^{\mathrm{d}}[l])\big\rangle_{\mathcal{H}_y} = f^{\mathrm{s}}_{y_0}(\mathbf{z}_i^{\mathrm{d}}[l]).
\end{aligned}
$$

Similarly, we have $f_{y_1}(\mathbf{z}_i^{\mathrm{d}}[l]) = f^{\mathrm{s}}_{y_1}(\mathbf{z}_i^{\mathrm{d}}[l])$, $f_w(\mathbf{z}_i^{\mathrm{d}}[l]) = f^{\mathrm{s}}_w(\mathbf{z}_i^{\mathrm{d}}[l])$, $f_{x,d}(\mathbf{z}_i^{l}) = f^{\mathrm{s}}_{x,d}(\mathbf{z}_i^{\mathrm{d}}[l])$, $f_{q_0,d}(y_i^{\mathrm{d}}, \mathbf{x}_i^{\mathrm{d}}) = f^{\mathrm{s}}_{q_0,d}(y_i^{\mathrm{d}}, \mathbf{x}_i^{\mathrm{d}})$, $f_{q_1,d}(y_i^{\mathrm{d}}, \mathbf{x}_i^{\mathrm{d}}) = f^{\mathrm{s}}_{q_1,d}(y_i^{\mathrm{d}}, \mathbf{x}_i^{\mathrm{d}})$. Hence,

$$
\widehat{\mathcal{L}}(f_{y_0}, f_{y_1}, f_{q_0}, f_{q_1}, f_x, f_w) = \widehat{\mathcal{L}}(f^{\mathrm{s}}_{y_0}, f^{\mathrm{s}}_{y_1}, f^{\mathrm{s}}_{q_0}, f^{\mathrm{s}}_{q_1}, f^{\mathrm{s}}_{x}, f^{\mathrm{s}}_{w}).
$$

The last equation implies that $\widehat{\mathcal{L}}(\cdot)$ depends only on the component of f_{y_0}, f_{y_1}, f_w, $f_{x,d}$, $f_{q_0,d}$, $f_{q_1,d}$ lying in the subspaces $\mathcal{U}_{y_0}$, $\mathcal{U}_{y_1}$, $\mathcal{U}_w$, $\mathcal{U}_x$, $\mathcal{U}_{q_0}$, $\mathcal{U}_{q_1}$, respectively. (2)

From (2) and (2), we obtain that each f_c is the weighted sum of elements in $\mathcal{U}_c$ ($c \in \{y_0, y_1, q_0, q_1, x, w\}$). This completes the proof. □

C PROOF OF LEMMA 2

PROOF. From Lemma 1, the objective function J is a combination of several components including $(\boldsymbol{\alpha}^c)^\top \mathrm{C}\boldsymbol{\alpha}^c$, $\mathbf{c}^\top\boldsymbol{\alpha}^c$, $-\mathbf{c}^\top \log\varphi(\mathbf{D}\boldsymbol{\alpha}^c)$, and $-(\mathbf{1}-\mathbf{w})^\top \log\varphi(-\mathbf{D}\boldsymbol{\alpha}^c)$, where $c \in \{y_0, y_1, w, x, q_0, q_1\}$, C is a positive semi-definite matrix, $\mathbf{c}$ is a vector, and $\mathbf{D}$ is a matrix computed by kernel functions.

For the first and second term, we have

$$
\nabla^2_{\boldsymbol{\alpha}^c}\big\{(\boldsymbol{\alpha}^c)^\top\mathrm{C}\boldsymbol{\alpha}^c\big\} = \mathrm{C} + \mathrm{C}^\top = 2\mathrm{C} \succeq 0, \quad \nabla^2_{\boldsymbol{\alpha}^c}\big\{\mathbf{c}^\top\boldsymbol{\alpha}^c\big\} = \mathbf{0} \succeq 0,
$$

where '$\succeq 0$' indicates that the matrix is positive semi-definite.

For the third term, we have

$$
\begin{aligned}
\nabla_{\boldsymbol{\alpha}^c}\big\{-\mathbf{w}^\top\log\varphi(\mathbf{D}\boldsymbol{\alpha}^c)\big\} &= -(\nabla_{\boldsymbol{\alpha}^c}\log\varphi(\mathbf{D}\boldsymbol{\alpha}^c))\mathbf{w}\\
&= -\mathbf{D}^\top\mathrm{diag}(\mathbf{w})\varphi(-\mathbf{D}\boldsymbol{\alpha}^c),
\end{aligned}
$$

and thus

$$
\begin{aligned}
\nabla^2_{\boldsymbol{\alpha}^c}\big\{-\mathbf{w}^\top\log\varphi(\mathbf{D}\boldsymbol{\alpha}^c)\big\} &= -(\nabla_{\boldsymbol{\alpha}^c}\varphi(-\mathbf{D}\boldsymbol{\alpha}^c))(\mathbf{D}^\top\mathrm{diag}(\mathbf{w}))^\top\\
&= -(\nabla_{\boldsymbol{\alpha}^c}\varphi(-\mathbf{D}\boldsymbol{\alpha}^c))\mathrm{diag}(\mathbf{w})\mathbf{D}\\
&= \mathbf{D}^\top\mathrm{diag}(\varphi(-\mathbf{D}\boldsymbol{\alpha}^c)\odot\varphi(\mathbf{D}\boldsymbol{\alpha}^c)\odot\mathbf{w})\mathbf{D} \succeq 0.
\end{aligned}
$$

Similarly, for the last term, we have

$$
\nabla_{\boldsymbol{\alpha}^c}\big\{-(\mathbf{1}-\mathbf{w})^\top\log\varphi(-\mathbf{D}\boldsymbol{\alpha}^c)\big\} = \mathbf{D}^\top\mathrm{diag}(\mathbf{1}-\mathbf{w})\varphi(\mathbf{D}\boldsymbol{\alpha}^c),
$$

and

$$
\begin{aligned}
&\nabla^2_{\boldsymbol{\alpha}^c}\big\{-(\mathbf{1}-\mathbf{w})^\top\log\varphi(-\mathbf{D}\boldsymbol{\alpha}^c)\big\}\\
&\quad= \mathbf{D}^\top\mathrm{diag}(\varphi(\mathbf{D}\boldsymbol{\alpha}^c)\odot\varphi(-\mathbf{D}\boldsymbol{\alpha}^c)\odot(\mathbf{1}-\mathbf{w}))\mathbf{D} \succeq 0.
\end{aligned}
$$

Consequently, J is convex because it is a linear combination of convex functions. □

REFERENCES

[1] Virginia Aglietti, Theodoros Damoulas, Mauricio Álvarez, and Javier González. 2020. Multi-task causal learning with gaussian processes. *Advances in Neural Information Processing Systems* 33 (2020), 6293–6304.

[2] Ahmed M Alaa and Mihaela van der Schaar. 2017. Bayesian inference of individualized treatment effects using multi-task gaussian processes. In *Advances in Neural Information Processing Systems*. 3424–3432.

[3] Elizabeth S Allman, Catherine Matias, John A Rhodes, et al. 2009. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics* 37, 6A (2009), 3099–3132.

[4] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. 2014. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research* 15, 1 (2014), 2773–2832.

[5] Elias Bareinboim, Sanghack Lee, Vasant Honavar, and Judea Pearl. 2013. Transportability from multiple environments with limited experiments. In *Advances in Neural Information Processing Systems*. 136–144.

[6] Elias Bareinboim and Judea Pearl. 2013. Causal transportability with limited experiments. In *Proceedings of the 27th AAAI conference on artificial intelligence*. 95–101.

[7] Elias Bareinboim and Judea Pearl. 2014. Transportability from multiple environments with limited experiments: Completeness results. In *Advances in neural information processing systems*. 280–288.

[8] Elias Bareinboim and Judea Pearl. 2016. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences* 113, 27 (2016), 7345–7352.

[9] Ioana Bica, Ahmed Alaa, and Mihaela Van Der Schaar. 2020. Time series deconfounder: Estimating treatment effects over time in the presence of hidden confounders. In *International Conference on Machine Learning*. PMLR, 884–895.

[10] Ioana Bica, Ahmed M Alaa, James Jordon, and Mihaela van der Schaar. 2020. Estimating counterfactual treatment outcomes over time through adversarially balanced representations. In *International Conference on Learning Representations*.

[11] Huigang Chen, Totte Harinen, Jeong-Yoon Lee, Mike Yung, and Zhenyu Zhao. 2020. CausalML: Python Package for Causal Machine Learning. arXiv:2002.11631 [cs.CY]

[12] Vincent Dorie. 2016. NPCI: Non-parametrics for causal inference.

[13] Donald P Green and Holger L Kern. 2012. Modeling heterogeneous treatment effects in survey experiments with Bayesian additive regression trees. *Public opinion quarterly* 76, 3 (2012), 491–511.

[14] Justin Grimmer, Solomon Messing, and Sean J Westwood. 2017. Estimating heterogeneous treatment effects and the effects of heterogeneous treatments with ensemble methods. *Political Analysis* 25, 4 (2017), 413–434.

[15] Jason Hartford, Greg Lewis, Kevin Leyton-Brown, and Matt Taddy. 2017. Deep IV: A flexible approach for counterfactual prediction. In *International Conference on Machine Learning*. 1414–1423.

[16] Nicholas C Henderson, Thomas A Louis, Chenguang Wang, and Ravi Varadhan. 2016. Bayesian analysis of heterogeneous treatment effects for patient-centered outcomes research. *Health Services and Outcomes Research Methodology* 16, 4 (2016), 213–233.

[17] Jennifer L Hill. 2011. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics* 20, 1 (2011), 217–240.

[18] Kosuke Imai and Marc Ratkovic. 2013. Estimating treatment effect heterogeneity in randomized program evaluation. *The Annals of Applied Statistics* 7, 1 (2013), 443–470.

[19] Nathan Kallus, Xiaojie Mao, and Angela Zhou. 2019. Interval estimation of individual-level causal effects under unobserved confounding. In *The 22nd International Conference on Artificial Intelligence and Statistics*. 2281–2290.

[20] Joseph B Kruskal. 1976. More factors than subjects, tests and treatments: an indeterminacy theorem for canonical decomposition and individual differences scaling. *Psychometrika* 41, 3 (1976), 281–293.

[21] Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. 2019. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences* 116, 10 (2019), 4156–4165.

[22] Manabu Kuroki and Judea Pearl. 2014. Measurement bias and effect restoration in causal inference. *Biometrika* 101, 2 (2014), 423–437.

[23] S. Lee, J. Correa, and E. Bareinboim. 2020. Generalized Transportability: Synthesis of Experiments from Heterogeneous Domains. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. AAAI Press, New York, NY.

[24] Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. 2017. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*. 6446–6456.

[25] Chaochao Lu, Bernhard Schölkopf, and José Miguel Hernández-Lobato. 2018. Deconfounding reinforcement learning in observational settings. *arXiv preprint arXiv:1812.10576* (2018).

[26] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. 2019. Fairness through Causal Awareness: Learning Causal Latent-Variable Models for Biased Data. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM, 349–358.

[27] Sara Magliacane, Thijs van Ommen, Tom Claassen, Stephan Bongers, Philip Versteeg, and Joris M Mooij. 2018. Domain adaptation by using causal inference to predict invariant conditional distributions. In *Advances in Neural Information Processing Systems*. 10846–10856.

[28] Microsoft Research. 2019. EconML: A Python Package for ML-Based Heterogeneous Treatment Effects Estimation. https://github.com/microsoft/EconML. Version 0.x.

[29] Mark R Montgomery, Michele Gragnolati, Kathleen A Burke, and Edmundo Paredes. 2000. Measuring living standards with proxy variables. *Demography* 37, 2 (2000), 155–174.

[30] Xinkun Nie and Stefan Wager. 2021. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika* 108, 2 (2021), 299–319.

[31] Miruna Oprescu, Vasilis Syrgkanis, and Zhiwei Steven Wu. 2019. Orthogonal random forest for causal inference. In *International Conference on Machine Learning*. PMLR, 4932–4941.

[32] Muhammad Osama, Dave Zachariah, and Thomas B Schön. 2019. Inferring heterogeneous causal effects in presence of spatial confounding. In *International Conference on Machine Learning*. PMLR, 4942–4950.

[33] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359.

[34] Judea Pearl. 1995. Causal diagrams for empirical research. *Biometrika* 82, 4 (1995), 669–688.

[35] Judea Pearl. 2000. *Causality: models, reasoning and inference*. Vol. 29. Springer.

[36] Judea Pearl and Elias Bareinboim. 2014. External validity: From do-calculus to transportability across populations. *Statist. Sci.* (2014), 579–595.

[37] Scott Powers, Junyang Qian, Kenneth Jung, Alejandro Schuler, Nigam H Shah, Trevor Hastie, and Robert Tibshirani. 2018. Some methods for heterogeneous treatment effect estimation in high dimensions. *Statistics in medicine* 37, 11 (2018), 1767–1787.

[38] Vineeth Rakesh, Ruocheng Guo, Raha Moraffah, Nitin Agarwal, and Huan Liu. 2018. Linked causal variational autoencoder for inferring paired spillover effects. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1679–1682.

[39] Chuan-Xian Ren, Xiao-Lin Xu, and Hong Yan. 2018. Generalized conditional domain adaptation: A causal perspective with low-rank translators. *IEEE transactions on cybernetics* 50, 2 (2018), 821–834.

[40] Stephanie K Riegg. 2008. Causal inference and omitted variable bias in financial aid research: Assessing solutions. *The Review of Higher Education* 31, 3 (2008), 329–354.

[41] Beatriz L Rodriguez, Darwin R Labarthe, Boji Huang, and Javier Lopez-Gomez. 1994. Rise of blood pressure with age. New evidence of population differences. *Hypertension* 24, 6 (1994), 779–785.

[42] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. 2018. Invariant models for causal transfer learning. *The Journal of Machine Learning Research* 19, 1 (2018), 1309–1342.

[43] Paul R Rosenbaum and Donald B Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70, 1 (1983), 41–55.

[44] Donald B Rubin. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology* 66, 5 (1974), 688.

[45] Donald B Rubin. 2005. Causal inference using potential outcomes: Design, modeling, decisions. *J. Amer. Statist. Assoc.* 100, 469 (2005), 322–331.

[46] Patrick Schwab, Lorenz Linhardt, and Walter Karlen. 2018. Perfect match: A simple method for learning representations for counterfactual inference with neural networks. *arXiv preprint arXiv:1810.00656* (2018).

[47] Uri Shalit, Fredrik D Johansson, and David Sontag. 2017. Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR.org, 3076–3085.

[48] Matt Taddy, Matt Gardner, Liyun Chen, and David Draper. 2016. A nonparametric bayesian analysis of heterogenous treatment effects in digital experimentation. *Journal of Business & Economic Statistics* 34, 4 (2016), 661–672.

[49] Takeshi Teshima, Issei Sato, and Masashi Sugiyama. 2020. Few-shot domain adaptation by causal mechanism transfer. In *International Conference on Machine Learning*. PMLR, 9458–9469.

[50] Victor Veitch, Yixin Wang, and David Blei. 2019. Using embeddings to correct for unobserved confounding in networks. In *Advances in Neural Information Processing Systems*. 13792–13802.

[51] Stefan Wager and Susan Athey. 2018. Estimation and inference of heterogeneous treatment effects using random forests. *J. Amer. Statist. Assoc.* 113, 523 (2018), 1228–1242.

[52] Sam Witty, Kenta Takatsu, David Jensen, and Vikash Mansinghka. 2020. Causal inference using Gaussian processes with structured latent confounders. In *International Conference on Machine Learning*. PMLR, 10313–10323.

[53] Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. 2018. Representation learning for treatment effect estimation from observational data. In *Advances in Neural Information Processing Systems*. 2633–2643.

[54] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. 2018. GANITE: Estimation of Individualized Treatment Effects using Generative Adversarial Nets. In *International Conference on Learning Representations*.

[55] Junzhe Zhang and Elias Bareinboim. 2017. Transfer learning in multi-armed bandit: a causal approach. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. 1778–1780.

[56] Yao Zhang, Alexis Bellot, and Mihaela van der Schaar. 2020. Learning overlapping representations for the estimation of individualized treatment effects. *arXiv preprint arXiv:2001.04754* (2020).