A Discussion on the Definition of (Causal) Foundation Models

In this paper, we focus on treatment effect estimation tasks (defined in Section 3.1). Our model is then tailored for generalizable zero-shot estimating average treatment effects. That is, given unseen datasets/contexts that contains observational records of covariates, treatments, and effects, we aim to estimate the underlying treatment effects using a forward pass of the underlying model.

This approach is inline with the definition of foundation models discussed in Bommasani et al. (2021): "any model that is trained on broad data (generally using self-supervision at scale) that can be adapted (e.g., fine-tuned) to a wide range of downstream tasks". Note that such *task-universality* of foundation models does not necessarily imply adaptability across different *machine learning formulations* (e.g., prediction, imputation, ATE, CATE, counterfactuals); instead, it can refer to adaptability across different *contexts* for a given task. This perspective is widely embraced by recent studies, such as those focusing on foundation models for tabular datasets (Zhang et al., 2023b), time series (Garza & Mergenthaler-Canseco, 2023; Das et al., 2023), and knowledge graphs (Galkin et al., 2023). These studies concentrate exclusively on a single type of task, but assess in-context generalization across datasets.

B Extended Related Works

As our work also intersects with the literature on neural causal estimation methods, we provide a discussion in this section.

Neural Estimation Methods for Treatment Effects. Research in this direction employs deep learning methods to estimate treatment effects, typically relying on standard assumptions that ensure identifiability, similar to our setting. A prominent approach focuses on learning a representation of the covariates that is predictive of the outcome (Johansson et al., 2016; Shalit et al., 2017; Yao et al., 2018). Following this, several methods have been proposed to combine outcome models learned through neural networks with balanced propensity weights (Alaa et al., 2017; Schwab et al., 2018; Du et al., 2021). Semi-parameteric estimation theory and doubly robust estimators have also been applied in neural estimation methods, e.g., using regularization (Shi et al., 2019) or shared representations (Chernozhukov et al., 2018). Another perspective of using neural network is to control for complex relationships and covariates. Kallus (2020a) extends adversarial covariate balancing (Kallus, 2020b) using flexible modeling with neural networks. Generative causal models have also been proposed to leverage the expressivity of neural networks to approximate structural causal models (Louizos et al., 2017; Kocaoglu et al., 2017; Alaa & Van Der Schaar, 2017; Yoon et al., 2018; Pawlowski et al., 2020; Xia et al., 2021; 2022), which then allows for the estimation of treatment effects. In addition, Xia et al. (2021) also proved that their proposed method can be used to test the identifiability of causal effect in terms of do-interventions (Pearl, 2009) in the general setting. Xia et al. (2022) extended such testing for counterfactual outcomes (Bareinboim et al., 2022). In (Melnychuk et al., 2022), the attention mechanism was employed to estimate treatment effect over time for a given unit. Concurrent to our work, Nilforoshan et al. (2023) proposed a meta-learning framework to learn causal effects of various structured treatments on the same population. Their method leverages information across different treatments, which allows for zero-shot learning on an unseen treatment. Our work can be viewed as orthogonal, as we focus on learning the causal effects of the same treatment across different populations.

C Omitted Proofs

C.1 Derivations of Eq. (1) and Eq. (2)

We first establish the conditional bias decomposition:

$$\begin{aligned}
&\mathbb{E}\left(\hat{\tau} - \tau_{SATE} \mid \{\boldsymbol{X}_i, T_i\}_{i=1}^N\right) \\
&= \mathbb{E}\left(\sum_{i=1}^N \alpha_i W_i Y_i - \sum_{i=1}^N \frac{1}{N}\big(Y_i(1) - Y_i(0)\big) \mid \{\boldsymbol{X}_i, T_i\}_{i=1}^N\right)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{N} \alpha_i W_i \mathbb{E}\left(Y_i(T_i) \mid \boldsymbol{X}_i, T_i\right) + \sum_{i=1}^{N} \frac{1}{N} \mathbb{E}\left(Y_i(1) - Y_i(0) \mid \boldsymbol{X}_i, T_i\right) \\
&= \sum_{i=1}^{N} \left(\alpha_i W_i \mathbb{E}\left(Y_i(0) \mid \boldsymbol{X}_i\right) + \alpha_i T_i \mathbb{E}\left(Y_i(1) - Y_i(0) \mid \boldsymbol{X}_i\right)\right) + \sum_{i=1}^{N} \frac{1}{N} \mathbb{E}\left(Y_i(1) - Y_i(0) \mid \boldsymbol{X}_i\right) \\
&= \sum_{i=1}^{N} (\alpha_i T_i - \frac{1}{N}) \mathbb{E}\left(Y_i(1) - Y_i(0) \mid \boldsymbol{X}_i\right) + \sum_{i=1}^{N} \alpha_i W_i \mathbb{E}\left(Y_i(0) \mid \boldsymbol{X}_i\right),
\end{aligned}
$$

where we use the assumption of consistency between observed and potential outcomes and non-interference between unit (SUTVA, Rubin (1990)) in the second equation and unconfoundedness in the third equation.

Formally, define a feature map $\phi : \mathbb{X} \to \mathcal{H}_\phi$, where $\mathbb{X}$ is the support of covariates and $\mathcal{H}_\phi$ is some Hilbert space. The unit-ball RKHS is given by $\mathcal{F}_\phi = \{f : \mathbb{X} \to \mathbb{R} \mid \exists \theta \in \mathcal{H}_\phi,\ s.t.\ f(x) = \langle \theta, \phi(x) \rangle,\ \forall x \in \mathbb{X}\ and\ \|\theta\| \leq 1\}$. Recall that $\langle \cdot, \cdot \rangle$ denotes the inner product of Hilbert space $\mathcal{H}_\phi$ and $\|\cdot\|$ denotes the associated norm. The adversarial upper bound of the square of the second term in the conditional bias can be calculated via

$$
\begin{aligned}
&\sup_{f \in \mathcal{F}_\phi} \left(\sum_{i=1}^{N} \alpha_i W_i f(\boldsymbol{X}_i) \right)^2 \\
&= \sup_{\theta \in \mathcal{H}_\phi, \|\theta\| \leq 1} \left(\sum_{i=1}^{N} \alpha_i W_i \Big\langle \theta, \phi(\boldsymbol{X}_i) \Big\rangle \right)^2 \\
&= \sup_{\theta \in \mathcal{H}_\phi, \|\theta\| \leq 1} \left(\Big\langle \theta, \sum_{i=1}^{N} \alpha_i W_i \phi(\boldsymbol{X}_i) \Big\rangle \right)^2 \\
&\leq \left\| \sum_{i=1}^{N} \alpha_i W_i \phi(\boldsymbol{X}_i) \right\|^2 = \boldsymbol{\alpha}^\top \boldsymbol{K}_\phi \boldsymbol{\alpha}.
\end{aligned}
$$

Recall that $[\boldsymbol{K}_\phi]_{ij} = W_i W_j \langle \phi(\boldsymbol{X}_i), \phi(\boldsymbol{X}_j) \rangle$. Therefore minimizing this adversarial loss subject to $\boldsymbol{\alpha} \in \mathbb{A}$ reduces to Eq. (1).

By evoking Theorem 1 in Tarr & Imai (2021), we have that Eq. (1) is equivalent to Eq. (2) for some $\lambda \geq 0$. However, the exact value of λ depends on $\boldsymbol{K}_\phi$. For example, if $\boldsymbol{K}_\phi$ is such that the minimum value of Eq. (1) is 0, then $\lambda = 0$. This is because the minimizer of Eq. (1) would also be the minimizer under the unnormalized constraint (Eq. (2) with $\lambda = 0$), as $\boldsymbol{\alpha}^\top \boldsymbol{K}_\phi \boldsymbol{\alpha} \geq 0$ for any $\boldsymbol{\alpha} \in \mathbb{R}^N$.

Conversely, we can also show that $\lambda > 0$ if $\boldsymbol{K}_\phi$ is of full rank.

Lemma 1. *If $\boldsymbol{K}_\phi$ if of full rank, then $\lambda > 0$.*

Proof. From the proof of Theorem 1 in Tarr & Imai (2021), we know that $\lambda = 0$ only if $q_* = \min_{\boldsymbol{W}^\top \boldsymbol{\alpha} = 0, \mathbf{0} \preceq \boldsymbol{\alpha} \preceq \mathbf{1}, \boldsymbol{\alpha} \neq \mathbf{0}} \frac{\sqrt{\boldsymbol{\alpha}^\top \boldsymbol{K}_\phi \boldsymbol{\alpha}}}{\mathbf{1}^\top \boldsymbol{\alpha}/2}$ is zero. However, since $\boldsymbol{K}_\phi$ is of full rank, it is positive definite. Thus for any $\boldsymbol{\alpha} \neq 0$, there is $\boldsymbol{\alpha}^\top \boldsymbol{K}_\phi \boldsymbol{\alpha} > 0$. Therefore $q_* > 0$. Consequently, $\lambda > 0$. □

C.2 Derivations of Eq. (3) and Eq. (4)

The dual form of Eq. (3) can be derived using its Lagrangian

$$
L(\boldsymbol{\beta}, \beta_0, \boldsymbol{\xi}, \boldsymbol{\alpha}, \bar{\boldsymbol{\alpha}}) = \frac{\lambda}{2} \|\boldsymbol{\beta}\|^2 + \sum_{i=1}^{N} \xi_i + \sum_{i=1}^{N} \alpha_i \left(1 - \xi_i - W_i \Big(\langle \boldsymbol{\beta}, \phi(\boldsymbol{X}_i) \rangle + \beta_0 \Big) \right) - \sum_{i=1}^{N} \bar{\alpha}_i \xi_i,
$$

where $\boldsymbol{\alpha} \succeq \mathbf{0}$ and $\bar{\boldsymbol{\alpha}} \succeq \mathbf{0}$. The primal form in Eq. (3) can be obtained by $\min_{\boldsymbol{\beta}, \beta_0, \xi_i} \max_{\boldsymbol{\alpha} \succeq \mathbf{0}, \bar{\boldsymbol{\alpha}} \succeq \mathbf{0}} L(\boldsymbol{\beta}, \beta_0, \boldsymbol{\xi}, \boldsymbol{\alpha}, \bar{\boldsymbol{\alpha}})$. If we exchange $\min\max$ with $\max\min$, solving

$\min_{\boldsymbol{\beta},\beta_0,\xi_i}$ by setting the derivatives to zero leads to

$$\nabla_{\boldsymbol{\beta}} L(\boldsymbol{\beta},\beta_0,\xi,\boldsymbol{\alpha},\bar{\boldsymbol{\alpha}}) = \lambda\boldsymbol{\beta} - \sum_{i=1}^{N} \alpha_i W_i \phi(\boldsymbol{X}_i) = \mathbf{0},$$
$$\nabla_{\beta_0} L(\boldsymbol{\beta},\beta_0,\xi,\boldsymbol{\alpha},\bar{\boldsymbol{\alpha}}) = -\sum_{i=1}^{N} \alpha_i W_i = 0,$$
$$\nabla_{\xi_i} L(\boldsymbol{\beta},\beta_0,\xi,\boldsymbol{\alpha},\bar{\boldsymbol{\alpha}}) = 1 - \alpha_i - \bar{\alpha}_i = 0,\ \forall\, i \in [N].$$

Plugging these in $L(\boldsymbol{\beta},\beta_0,\xi,\boldsymbol{\alpha},\bar{\boldsymbol{\alpha}})$, we can reduce $\max_{\boldsymbol{\alpha}\succeq\mathbf{0},\bar{\boldsymbol{\alpha}}\succeq\mathbf{0}} \min_{\boldsymbol{\beta},\beta_0,\xi_i} L(\boldsymbol{\beta},\beta_0,\boldsymbol{\xi},\boldsymbol{\alpha},\bar{\boldsymbol{\alpha}})$ to Eq. (2). Thus it is the dual form of Eq. (3).

In addition, we can also derive Eq. (4). It is easy to check that Slater's condition holds for the primal SVM problem in Eq. (3). Thus it satisfies strong duality. Therefore any optimal solutions to the primal-dual problems must satisfy the KKT condition $\lambda\boldsymbol{\beta}^* = \sum_{j=1}^{N} \alpha_j^* W_j \phi(\boldsymbol{X}_j)$.

C.3 Derivations of Eq. (6)

From the Taylor expansion

$$\begin{aligned}\exp(\boldsymbol{k}_i^\top \boldsymbol{k}_j/\sqrt{D}) &= \sum_{l=0}^{+\infty} \frac{1}{l!} (\boldsymbol{k}_i^\top \boldsymbol{k}_j/\sqrt{D})^l \\ &= \sum_{l=0}^{+\infty} \sum_{N_1+\ldots+N_D=l} \frac{\big([\boldsymbol{k}_i]_1^{N_1}\ldots[\boldsymbol{k}_i]_D^{N_D}\big)\big([\boldsymbol{k}_j]_1^{N_1}\ldots[\boldsymbol{k}_j]_D^{N_D}\big)}{D^{l/2} N_1!\ldots N_D!},\end{aligned}$$

we have that $\exp(\boldsymbol{k}_i^\top \boldsymbol{k}_j/\sqrt{D}) = \langle \phi(\boldsymbol{X}_i), \phi(\boldsymbol{X}_j)\rangle$ if

$$\phi(\boldsymbol{x}) = \left(\frac{[\boldsymbol{k}]_1^{N_1}\ldots[\boldsymbol{k}]_D^{N_D}}{D^{l/2}(N_1!\ldots N_D!)^{1/2}} \right)_{N_1+\ldots+N_D=l,\ l\in\mathbb{N}}. \tag{9}$$

Here $\boldsymbol{k}$ denotes the key embedding of $\boldsymbol{x}$ following the same transformation that $\boldsymbol{k}_i$ is obtained from $\boldsymbol{X}_i$. Note that we allow the transformation to depend on $\boldsymbol{X}$, which corresponds to a data-dependent kernel.

Using this expression, the i-th output of the self-attention layer when $\boldsymbol{Q} = \boldsymbol{K}$ can be equivalently written as

$$\sum_{j=1}^{N} \frac{\exp\big(\boldsymbol{k}_i^\top \boldsymbol{k}_j/\sqrt{D}\big)}{\sum_{j'=1}^{N} \exp\big(\boldsymbol{k}_i^\top \boldsymbol{k}_{j'}/\sqrt{D}\big)} v_j = \sum_{j=1}^{N} \frac{\langle \phi(\boldsymbol{X}_i), \phi(\boldsymbol{X}_j)\rangle}{h(\boldsymbol{X}_i)} v_i = \sum_{j=1}^{N} \frac{v_j}{h(\boldsymbol{X}_j)} \langle \phi(\boldsymbol{X}_j), \phi(\boldsymbol{X}_i)\rangle.$$

C.4 Proof of Theorem 1

We first state its formal version:

Theorem 1. *If the covariates $\boldsymbol{X}$ satisfy that $\phi(\boldsymbol{X}_1), ..., \phi(\boldsymbol{X}_N)$ are linearly independent, then Algorithm 1 recovers the optimal balancing weight at the global minimum of the penalized hinge loss in Eq. (7).*

In particular, the optimal solution $\boldsymbol{\alpha}^*$ to Eq. (1), in which the feature function ϕ is defined using the optimal neural network parameters via Eq. (9), can be obtained using the optimal neural network parameters that minimize Eq. (7) via $\alpha_j^* = \lambda v_j / h(\boldsymbol{X}_j) W_j$.

Proof. Denote $\boldsymbol{\beta} = \sum_{j=1}^{N} \frac{v_j}{h(\boldsymbol{X}_j)} \phi(\boldsymbol{X}_j)$, then using Eq. (6), we can rewrite the loss function in Eq. (7) as

$$\mathcal{L}_{\boldsymbol{\theta}}(\mathbb{D}) = \frac{\lambda}{2}\|\boldsymbol{\beta}\|^2 + \sum_{i=1}^{N} \big[1 - W_i\big(\langle \boldsymbol{\beta}, \phi(\boldsymbol{X}_i)\rangle + \beta_0\big)\big]_+.$$

Denote $\xi_i = \big[1 - W_i\big(\langle \boldsymbol{\beta}, \phi(\boldsymbol{X}_i)\rangle + \beta_0\big)\big]_+$, then minimizing $\mathcal{L}_{\boldsymbol{\theta}}(\mathbb{D})$ can be equivalently written as

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \frac{\lambda}{2}\|\boldsymbol{\beta}\|^2 + \sum_{i=1}^N \xi_i, \\ s.t. \quad & W_i\left(\big\langle \boldsymbol{\beta}, \phi(\boldsymbol{X}_i)\big\rangle + \beta_0\right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i \in [N]. \end{aligned}$$

Thus at the optimal $\boldsymbol{\theta}$, the corresponding $\boldsymbol{\beta}$ is also the optimal solution to

$$\begin{aligned} \min_{\boldsymbol{\beta}, \beta_0, \boldsymbol{\xi}} \quad & \frac{\lambda}{2}\|\boldsymbol{\beta}\|^2 + \sum_{i=1}^N \xi_i, \\ s.t. \quad & W_i\left(\big\langle \boldsymbol{\beta}, \phi(\boldsymbol{X}_i)\big\rangle + \beta_0\right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i \in [N], \end{aligned}$$

where ϕ is defined using the optimal $\boldsymbol{\theta}$. This recovers the primal SVM problem. By the primal-dual connection proven in Appendix C.2, if we denote the optimal solution to the dual problem (which is Eq. (2)) as $\boldsymbol{\alpha}^*$, we have

$$\lambda \boldsymbol{\beta} = \sum_{j=1}^N \alpha_j^* W_j \phi(\boldsymbol{X}_j).$$

Consequently, by the definition of $\boldsymbol{\beta}$, we have

$$\sum_{j=1}^N \frac{\lambda v_j}{h(\boldsymbol{X}_j)} \phi(\boldsymbol{X}_j) = \sum_{j=1}^N \alpha_j^* W_j \phi(\boldsymbol{X}_j).$$

By the assumption that $\phi(\boldsymbol{X}_1), ..., \phi(\boldsymbol{X}_N)$ are linearly independent, we must have $\frac{\lambda v_j}{h(\boldsymbol{X}_j)} = \alpha_j^* W_j$ for all $j \in [N]$. Therefore $\alpha_j^* = \lambda v_j / h(\boldsymbol{X}_j) W_j$. □

Remark 1. *Note that when $\phi(\boldsymbol{X}_1), ..., \phi(\boldsymbol{X}_N)$ are linearly independent, the matrix $\boldsymbol{K}_\phi = [W_1\phi(\boldsymbol{X}_1), ..., W_N\phi(\boldsymbol{X}_N)]^\top [W_1\phi(\boldsymbol{X}_1), ..., W_N\phi(\boldsymbol{X}_N)]$ is of full rank. Thus by Lemma 1, there is $\lambda > 0$. Conversely, using a similar decomposition, we know that if $\hat{\boldsymbol{K}}_\phi = [\phi(\boldsymbol{X}_1), ..., \phi(\boldsymbol{X}_N)]^\top [\phi(\boldsymbol{X}_1), ..., \phi(\boldsymbol{X}_N)]$ is of full rank, then $\phi(\boldsymbol{X}_1), ..., \phi(\boldsymbol{X}_N)$ are linearly independent. Since $\hat{\boldsymbol{K}}_\phi = \exp(\boldsymbol{K}\boldsymbol{K}^\top/\sqrt{D})$, we have $\phi(\boldsymbol{X}_1), ..., \phi(\boldsymbol{X}_N)$ linearly independent if $\boldsymbol{K}$ is of row rank N. Thus the assumption on $\boldsymbol{X}$ in Theorem 1 is satisfied when $\boldsymbol{K}$ is of row rank N.*

D ALTERNATIVE OBJECTIVES

Consider minimizing the square of both terms in the conditional bias, which we decompose into the following form

$$\begin{aligned} & \left(\mathbb{E}\left(\hat{\tau} - \tau_{SATE} \mid \{\boldsymbol{X}_i, T_i\}_{i=1}^N\right)\right)^2 \\ = & \left(\sum_{i=1}^N \alpha_i W_i \mathbb{E}\big(Y_i(T_i)|\boldsymbol{X}_i, T_i\big) - \frac{1}{N}\sum_{i=1}^N \Big(\mathbb{E}\big(Y_i(1)|\boldsymbol{X}_i\big) - \mathbb{E}\big(Y_i(0)|\boldsymbol{X}_i\big)\Big)\right)^2. \end{aligned} \tag{10}$$

Denote the outcome models $\mathbb{E}(Y_i(1)|\boldsymbol{X}_i) = f_1(\boldsymbol{X}_i)$ and $\mathbb{E}(Y_i(0)|\boldsymbol{X}_i) = f_0(\boldsymbol{X}_i)$. We choose to minimize the above term in worst case over all possible potential outcome models $(f_0, f_1) \in \mathcal{F}_\phi^2$. Here the space $\mathcal{F}_\phi^2$ is defined as $\mathcal{F}_\phi^2 = \{(f_0, f_1) \mid f_0 \in \mathcal{F}_\phi, f_1 \in \mathcal{F}_\phi\}$.

Suppose $f_0(x) = \langle \phi(x), \theta_0 \rangle$ and $f_1(x) = \langle \phi(x), \theta_1 \rangle$ for $\theta_0, \theta_1 \in \mathcal{H}_\phi, \|\theta_0\| \leq 1, \|\theta_1\| \leq 1$. We can bound Eq. (10) with respect to all outcome models in $\mathcal{F}_\phi^2$ as

$$
\begin{aligned}
&\left(\sum_{i=1}^N \alpha_i W_i f_{T_i}(\boldsymbol{X}_i) - \frac{1}{N} \sum_{i=1}^N \big(f_1(\boldsymbol{X}_i) - f_0(\boldsymbol{X}_i) \big) \right)^2 \\
&= \left(\left\langle \sum_{i \in \mathbb{T}} \alpha_i W_i \phi(\boldsymbol{X}_i) - \frac{1}{N} \sum_{i \in [N]} \phi(\boldsymbol{X}_i), \theta_1 \right\rangle + \left\langle \sum_{i \in \mathbb{C}} \alpha_i W_i \phi(\boldsymbol{X}_i) + \frac{1}{N} \sum_{i \in [N]} \phi(X_i), \theta_0 \right\rangle \right)^2 \\
&\leq 2 \left(\sum_{i \in \mathbb{T}} \alpha_i W_i \phi(\boldsymbol{X}_i) - \frac{1}{N} \sum_{i \in [N]} \phi(\boldsymbol{X}_i) \right)^2 + 2 \left(\sum_{i \in \mathbb{C}} \alpha_i W_i \phi(\boldsymbol{X}_i) + \frac{1}{N} \sum_{i \in [N]} \phi(\boldsymbol{X}_i) \right)^2
\end{aligned}
$$

where the inequality uses Cauchy-Schwartz inequality. Minimizing this upper bound subject to $\boldsymbol{\alpha} \in \mathbb{A}$ is equivalent to solving

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}^\top \boldsymbol{G}_\phi \boldsymbol{\alpha} + \boldsymbol{\alpha}^\top \boldsymbol{g}_\phi, \\
s.t. \quad & \sum_{i \in \mathbb{T}} \alpha_i = \sum_{i \in \mathbb{C}} \alpha_i = 1, \quad \mathbf{0} \preceq \boldsymbol{\alpha} \preceq \mathbf{1}.
\end{aligned} \tag{11}
$$

Here

$$
\begin{aligned}
[\boldsymbol{G}_\phi]_{i,j} &= \delta_{W_i = W_j} \langle \phi(\boldsymbol{X}_i), \phi(\boldsymbol{X}_j) \rangle, \\
[\boldsymbol{g}_\phi]_i &= -\frac{2}{N} \sum_{j=1}^N \langle \phi(\boldsymbol{X}_i), \phi(\boldsymbol{X}_i) \rangle.
\end{aligned}
$$

It is easy to show that $\boldsymbol{G}_\phi \succeq 0$ as it can be decomposed into two submatrixes which are positive semi-definite. In addition, as $\langle \phi(\boldsymbol{X}_i), \phi(\boldsymbol{X}_j) \rangle = \exp(\boldsymbol{k}_i^\top \boldsymbol{k}_j / \sqrt{D}) > 0$, we know that $\boldsymbol{g}_\phi \prec \mathbf{0}$.

To come up with a consistent gradient-based solver, notice first that Eq. (11) is equivalent to the following unnormalized problem for some $\lambda, \mu \geq 0$

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}^\top \boldsymbol{G}_\phi \boldsymbol{\alpha} + 2\mu \cdot \boldsymbol{g}_\phi^\top \boldsymbol{\alpha} - 2\lambda \cdot \mathbf{1}^\top \boldsymbol{\alpha}, \\
s.t. \quad & \boldsymbol{W}^\top \boldsymbol{\alpha} = 0, \quad \mathbf{0} \preceq \boldsymbol{\alpha} \preceq \mathbf{1}.
\end{aligned} \tag{12}
$$

This can be shown similarly to the proof of Theorem 1 in Tarr & Imai (2021). We escape the details but provide the following main steps:

1. We first show that for some $\epsilon_\lambda, \epsilon_\mu \geq 0$, Eq. (12) is equivalent to

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}^\top \boldsymbol{G}_\phi \boldsymbol{\alpha}, \\
s.t. \quad & \boldsymbol{W}^\top \boldsymbol{\alpha} = 0, \quad \mathbf{0} \preceq \boldsymbol{\alpha} \preceq \mathbf{1}, \quad -\boldsymbol{g}_\phi^\top \boldsymbol{\alpha} \geq \epsilon_\mu, \quad \mathbf{1}^\top \boldsymbol{\alpha} \geq \epsilon_\lambda.
\end{aligned}
$$

2. Next, we show that the above problem is equivalent to

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \sqrt{\boldsymbol{\alpha}^\top \boldsymbol{G}_\phi \boldsymbol{\alpha}}, \\
s.t. \quad & \boldsymbol{W}^\top \boldsymbol{\alpha} = 0, \quad \mathbf{0} \preceq \boldsymbol{\alpha} \preceq \mathbf{1}, \quad -\boldsymbol{g}_\phi^\top \boldsymbol{\alpha} \geq \epsilon_\mu, \quad \mathbf{1}^\top \boldsymbol{\alpha} \geq \epsilon_\lambda,
\end{aligned}
$$

   which is equivalent to

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \sqrt{\boldsymbol{\alpha}^\top \boldsymbol{G}_\phi \boldsymbol{\alpha}} + \nu_\mu \cdot \boldsymbol{g}_\phi^\top \boldsymbol{\alpha} - \nu_\lambda \mathbf{1}^\top \alpha, \\
s.t. \quad & \boldsymbol{W}^\top \boldsymbol{\alpha} = 0, \quad \mathbf{0} \preceq \boldsymbol{\alpha} \preceq \mathbf{1}.
\end{aligned}
$$

   for some $\nu_\lambda, \nu_\mu \geq 0$.

3. For some $\lambda \geq 0$, the above problem is equivalent to

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{\sqrt{\boldsymbol{\alpha}^\top \boldsymbol{G}_\phi \boldsymbol{\alpha}} + \nu_\mu \cdot \boldsymbol{g}_\phi^\top \boldsymbol{\alpha}}{\mathbf{1}^\top \boldsymbol{\alpha}}, \\ s.t. \quad & \boldsymbol{W}^\top \boldsymbol{\alpha} = 0, \quad \mathbf{0} \preceq \boldsymbol{\alpha} \preceq \mathbf{1}. \end{aligned}$$

Since this problem is scale-free, it is equivalent to

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{\sqrt{\boldsymbol{\alpha}^\top \boldsymbol{G}_\phi \boldsymbol{\alpha}} + \nu_\mu \cdot \boldsymbol{g}_\phi^\top \boldsymbol{\alpha}}{\mathbf{1}^\top \boldsymbol{\alpha}}, \\ s.t. \quad & \sum_{i \in \mathbb{T}} \alpha_i = \sum_{i \in \mathbb{C}} \alpha_i = 1, \quad \mathbf{0} \preceq \boldsymbol{\alpha} \preceq \mathbf{1}, \end{aligned}$$

i.e.,

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \sqrt{\boldsymbol{\alpha}^\top \boldsymbol{G}_\phi \boldsymbol{\alpha}} + \nu_\mu \cdot \boldsymbol{g}_\phi^\top \boldsymbol{\alpha}, \\ s.t. \quad & \sum_{i \in \mathbb{T}} \alpha_i = \sum_{i \in \mathbb{C}} \alpha_i = 1, \quad \mathbf{0} \preceq \boldsymbol{\alpha} \preceq \mathbf{1}, \end{aligned}$$

4. Using similar arguments as above, one can show the above problem is equivalent to

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}^\top \boldsymbol{G}_\phi \boldsymbol{\alpha} + \boldsymbol{g}_\phi^\top \boldsymbol{\alpha}, \\ s.t. \quad & \sum_{i \in \mathbb{T}} \alpha_i = \sum_{i \in \mathbb{C}} \alpha_i = 1, \quad \mathbf{0} \preceq \boldsymbol{\alpha} \preceq \mathbf{1}, \end{aligned}$$

for some $\mu \geq 0$.

The primal form of Eq. (12) can be written as

$$\begin{aligned} \min_{\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \beta_0, \boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{\beta}_1\|^2 + \frac{1}{2}\|\boldsymbol{\beta}_2\|^2 + \sum_{i=1}^N \xi_i, \\ s.t. \quad & \big(\langle \boldsymbol{\beta}_1, \phi(\boldsymbol{X}_i)\rangle + \beta_0\big) \geq \lambda - \mu[\boldsymbol{g}_\phi]_i - \xi_i, \quad \forall i \in \mathbb{T} \\ & \big(\langle \boldsymbol{\beta}_2, \phi(\boldsymbol{X}_i)\rangle - \beta_0\big) \geq \lambda - \mu[\boldsymbol{g}_\phi]_i - \xi_i, \quad \forall i \in \mathbb{C} \\ & \xi_i \geq 0, \quad \forall i \in [N]. \end{aligned}$$

Following similar derivations in Appendix C, we can write out an unconstrained loss function

$$\begin{aligned} \mathcal{L}_{\boldsymbol{\theta}}(\mathbb{D}) = & \frac{1}{2}\left\| \sum_{j \in \mathbb{T}} \frac{v_j}{h(\boldsymbol{X}_j)} \phi(\boldsymbol{X}_j) \right\|^2 + \frac{1}{2}\left\| \sum_{j \in \mathbb{C}} \frac{v_j}{h(\boldsymbol{X}_j)} \phi(\boldsymbol{X}_j) \right\|^2 \\ & + \Big[\lambda - \mu[\boldsymbol{g}_\phi]_{\mathbb{T}} - \big(\mathrm{softmax}(\boldsymbol{K}_{\mathbb{T}} \boldsymbol{K}_{\mathbb{T}}^\top / \sqrt{D}) \boldsymbol{V}_{\mathbb{T}} + \beta_0\big)\Big]_+ \\ & + \Big[\lambda - \mu[\boldsymbol{g}_\phi]_{\mathbb{C}} - \big(\mathrm{softmax}(\boldsymbol{K}_{\mathbb{C}} \boldsymbol{K}_{\mathbb{C}}^\top / \sqrt{D}) \boldsymbol{V}_{\mathbb{C}} - \beta_0\big)\Big]_+, \end{aligned}$$

where the optimal $\boldsymbol{\alpha}^*$ solving Eq. (11) can be read off as $\alpha_i = \frac{v_i}{h(\boldsymbol{X}_i)}$.

For the conditional mean square error, under regularity constraints in Bennett & Kallus (2019), we can also use the same upper bound as above (up to an additive $\mathcal{O}(1/N)$ gap). Therefore the same derivation holds. However, as this loss function separates the treated group from the control group aside from sharing the constant intercept β_0, it might not be preferable than the objective proposed in the main text.

E NON-BINARY TREATMENTS

Consider a generalization to the setting in Section 3.1, where the dataset $\mathbb{D} = \{(\boldsymbol{X}_i, \boldsymbol{T}_i, Y_i)\}_{i \in [N]}$ in which $\boldsymbol{T}_i$ is a S-dimensional vector of multiple binary treatments. Let $Y_i^s(t)$ be the potential outcome of assigning treatment $[\boldsymbol{T}_i]_s = t$.

Assuming SUTVA ($Y_i = Y_i^s([\boldsymbol{T}_i]_s)$) and unconfoundedness. Denote $\mathbb{T}^s = \{i \in [N] : [\boldsymbol{T}_i]_s = 1\}$ and $\mathbb{C}^s = \{i \in [N] : [\boldsymbol{T}_i]_s = 0\}$. We consider weighted estimators in the form of

$$\hat{\tau}^s = \sum_{i \in \mathbb{T}^s} \alpha_i Y_i^s(1) - \sum_{i \in \mathbb{C}^s} \alpha_i Y_i^s(0)$$

for the sample average treatment of the s-th treatment

$$\tau^s_{SATE} = \frac{1}{N} \sum_{i=1}^{N} \big(Y_i^s(1) - Y_i^s(0)\big).$$

Following the same derivations in Section 3 and Appendix C, we can obtain a dual-SVM formulation to optimize $\boldsymbol{\alpha}$ in the adversarial case. This dual-SVM formulation can then be transformed into its primal problem. As self-attention is implicitly implementing the predictor in the primal problem, we can then read off the optimal $\boldsymbol{\alpha}^*$ by training this self-attention-based neural network with a penalized hinge loss.

However, as we would like to evaluate the sample average treatment for multiple treatments, we can actually aggregate S SVM problems together using the flexibility of self-attention layers. Namely, instead of consider a one-dimensional value vector $\boldsymbol{V}$ in Section 3.2, we use $\boldsymbol{V} \in \mathbb{R}^{N \times S}$, where the s-th dimension corresponds to the s-th treatment. By minimizing the following loss function

$$\mathcal{L}_{\boldsymbol{\theta}}(\mathbb{D}) = \frac{\lambda}{2} \sum_{s=1}^{S} \left\| \sum_{j=1}^{N} \frac{[\boldsymbol{V}]_{js}}{h(\boldsymbol{X}_j)} \phi(\boldsymbol{X}_j) \right\|^2 + \sum_{s=1}^{S} \Big[\mathbf{1} - \boldsymbol{W}_{:,s} \big(\mathrm{softmax}(\boldsymbol{K}\boldsymbol{K}^\top / \sqrt{D}) \boldsymbol{V}_{:,s} + \beta_0 \big) \Big]_+,$$

we can read off the optimal balancing weight $\boldsymbol{\alpha}$ for the s-th treatment via $\lambda \cdot \boldsymbol{V}_{:,s} / h(\boldsymbol{X}) \boldsymbol{W}_{:,s}$

F Individual Treatment Effect Estimation

In this section, we further consider the problem of estimating individual treatment effect (ITE) in the binary treatment setup of Section 3. Here we present one possible algorithmic approach to approximate ITEs with CInA. Without loss of generality, suppose $T_1 = 1$ and we would like to estimate ITE on the first unit $\mathbb{E}(Y_1(1) - Y_1(0) \mid \boldsymbol{X}_1)$.

Denote the "counterfactual dataset" by replacing the first sample with $(\boldsymbol{X}_1, 0, \hat{Y}_1(0))$ as $\hat{\mathbb{D}}$, where $\hat{Y}_1(0)$ is a realization of $Y_1(0)$. Note that we do not have access to the value of $\hat{Y}_1(0)$. However, we do have access to the covariates and treatments of $\hat{\mathbb{D}}$. As these are all the required inputs to Algorithm 1, we can compute the optimal balancing weight for this counterfactual dataset $\hat{\mathbb{D}}$, which we denote as $\hat{\boldsymbol{\alpha}}$.

Notice that the sample average treatments of $\mathbb{D}$ are $\hat{\mathbb{D}}$ should be the same, as they are defined for the same set of units. Therefore the two weighted estimators are approximating the same τ_{SATE} (or ATE when N increases) and thus

$$\begin{aligned} &\sum_{i \in \mathbb{T}} \alpha_i \mathbb{E}(Y_i(1) \mid \boldsymbol{X}_i) - \sum_{i \in \mathbb{C}} \alpha_i \mathbb{E}(Y_i(0) \mid \boldsymbol{X}_i) \\ \approx &\sum_{i \in \mathbb{T} \setminus \{1\}} \hat{\alpha}_i \mathbb{E}(Y_i(1) \mid \boldsymbol{X}_i) - \sum_{i \in \mathbb{C}} \hat{\alpha}_i \mathbb{E}(Y_i(0) \mid \boldsymbol{X}_i) - \hat{\alpha}_0 \mathbb{E}(\hat{Y}_1(0) \mid \boldsymbol{X}_1). \end{aligned}$$

Therefore we have the following approximation

$$\hat{\alpha}_1 \mathbb{E}(\hat{Y}_1(0) \mid \boldsymbol{X}_1) \approx -\alpha_1 Y_1(1) + \sum_{i \in \mathbb{T} \setminus \{1\}} (\hat{\alpha}_i - \alpha_i) Y_i(1) - \sum_{i \in \mathbb{C}} (\hat{\alpha}_i - \alpha_i) Y_i(0).$$

As we have access to all individual terms on the right, we can compute an approximation of $\mathbb{E}(Y_1(0) \mid \boldsymbol{X}_1)$, using this formula as long as $\hat{\alpha}_0 \neq 0$.[2]

[2]Once we have these estimands, policy evaluation can done via plug-in estimations.

To enhance the robustness of this estimation, we can also compute this for units with covariates closed to $\boldsymbol{X}_1$, e.g., using KNNs (Devroye et al., 1994; Li & Tran, 2009), which would give consistent estimations for conditional expectations. Algorithm 4 summarizes this procedure, where Algorithm 3 can be used instead of Algoritm 1 to estimate ITE in a zero-shot fashion.

Algorithm 4 CInA for ITE.

Input: Covariates $\boldsymbol{X}$ and treatments $\boldsymbol{W}$.
Output: Estimation of $\mathbb{E}(Y_1(1) - Y_1(0) \mid \boldsymbol{X}_1)$.
Hyper-parameter: penalty weight $\lambda > 0$.
Initialize $\tau = \varnothing$.
for unit i with $\boldsymbol{X}_i \approx \boldsymbol{X}_1$ **do**
    Run Algorithm 1 on $\boldsymbol{X}, \boldsymbol{W}$ to obtain $\boldsymbol{\alpha}$.
    Set $\hat{\boldsymbol{W}}$ to be $\boldsymbol{W}$ except $\hat{W}_i = -W_i$.
    Run Algorithm 1 on $\boldsymbol{X}, \hat{\boldsymbol{W}}$ to obtain $\hat{\boldsymbol{\alpha}}$.
    Let $\hat{\alpha}_i \mathbb{E}(\hat{Y}_i(1 - T_i) \mid \boldsymbol{X}_i) = -\alpha_i Y_i(T_i) + \sum_{j \neq i, T_j = T_i} (\hat{\alpha}_j - \alpha_j) Y_j(T_j) - \sum_{T_j \neq T_i} (\hat{\alpha}_j - \alpha_j) Y_j(T_j)$.
    Append $W_i \cdot (\mathbb{E}(\hat{Y}_i(1 - T_i) \mid \boldsymbol{X}_i) - Y_i(T_i))$ to τ if $\hat{\alpha}_i \neq 0$.
return Average of τ.

G Dataset Details

The details of the datasets for simulation A are provided in Section 5.1. We now provide the details of ER-5000 and the real-world datasets. Code for downloading and pre-processing these datasets will be provided upon publication.

ER-5000. Each of the ER-5000 datasets is generated following the structural causal model (SCM) framework. The detailed procedure is as follows. First, we sample a random directed acyclic graph (DAG) from the Erdős-Rényi random graph model (Erdős & Rényi, 1960) with edge probability sampled from 0.25 to 0.5. Then, Based on the sampled DAG, we sample the corresponding functional relationships using a linear weight sampler, with random weights sampled from a uniform distribution between 0 and 3. Next, a treatment node and effect node is randomly chosen. For each non-treatment node, we use additive gaussian random noise with standard deviation randomly sampled uniformly between 0.2 and 2. For treatment node, we specify a Bernoulli distribution with logit equal to the functional output of the corresponding node. Finally, we simulate each variable (in $\boldsymbol{X}$, T and Y) using the sampled DAG, functional relationships, and noises.

IHDP and IHDP-resampled. The Infant Health and Development Program (IHDP) dataset is a semi-dataset complied by Hill (2011). We use the existing versions from Chernozhukov et al. (2022), which are sampled using the outcome model implemented as setting A in (Dorie, 2016). Each dataset comprises of 747 units and 25 covaraites measuring the aspects of children and their mothers. For IHDP, the treatment group (139 out of 747 units) has been made imbalanced by removing a biased subset of the treated population. A total of 1000 datasets are used (following Shi et al. (2019)), where different datasets only differ in terms of outcome values. For IHDP-resampled, 100 datasets are used where the treatments are resampled by setting the propensity score to "True" in the (Dorie, 2016).

Twins. Introduced by Louizos et al. (2017), this is a semi-synthetic dataset based on the real data on twin births and twin mortality rates in the US from 1989 to 1991 (Almond et al., 2005). The treatment is "born the heavier twin", which is simulated as a function of the GESTAT10 covariates. Therefore this dataset is confounded. After assigning the treatment for each pair of twins, the dataset is constructed by hiding the other twin. We downloaded the dataset and processed it following Neal et al. (2020).

LaLonde CPS and PSID. We also use the datasets from LaLonde (1986), in which the treatment is job training and the outcomes are income and employment status after training. The ground-truth average treatment effect is computed using a randomized study, where we use the observational data to estimate it. The observational data has multiple versions. We use both the PSID-1 and CPS-1 versions for our experiments (Dehejia & Wahba, 1999).

ACIC. The data for the 2018 Atlantic Causal Inference Conference competition (ACIC) (Shimoni et al., 2018) comprises of serveral semi-synthetic datasets derived from the linked birth and infant death (LBIDD) data (MacDorman & Atkinson, 1998). The data-generating process is described in (Shimoni et al., 2018). In our experiment, we use datasets containing $1k$ or $10k$ samples.[3] In the experiments in Section 5, a total of 293 datasets (each of size $1k$) were used, where 93 were left out for testing. In Appendix I, we extend this to datasets of size $10k$, where a total of 288 datasets were used and 88 among these were left out for testing. We use datasets with polynomial link function for training and validation. For testing, we use datasets with exponential link functions thus creating a harder task for evaluating our methods.

H Implementation Details

Code for our method will be released on GitHub upon publication. Below we describe the architecture, hyper-parameters, training procedures and other details of our method. We also provide the implementation details of the baselines. Finally, we discuss a new data augmentation technique that we observe to be helpful on certain datasets.

H.1 CINA

Pre-processing and Padding. For Algorithm 2, we might encounter multiple datasets with different number of samples. We wish them to share the same transformation from $\boldsymbol{W}, \boldsymbol{K}$ to $\boldsymbol{V} \in \mathbb{R}^{N\times 1}$, where N is the number of units in the corresponding dataset. For this, we adopt similar pre-processing steps as in natural language. We pad all datasets to the same size (i.e., adding dumy units to smaller datasets) and save the masks that indicate these paddings. During back-propagation, we use this mask to make sure that the loss function is only computed using actual units.

Model Configurations. We describe the architecture used in Algorithm 2, as the single-dataset version uses the same components aside from parametrizing the values $\boldsymbol{V}$ directly as learnable parameters. An illustration of the forward pass is provided in Figure 2.

For the transformation from covariates $\boldsymbol{X}$ to keys $\boldsymbol{K}$, we implemented two versions: (1) an identical mapping followed by a batch-norm layer $\boldsymbol{K} = \mathrm{bn}(\boldsymbol{X})$, (2) a projected mapping followed by a batch-norm layer $\boldsymbol{k}_i = \mathrm{bn} \circ \mathrm{relu} \circ \mathrm{linear}(\boldsymbol{X}_i)$. In our first simulation study in Section 5.1, we observe that the projection to be marginally helpful and thus report all the results based on the identical mapping.

For the transformation from $\boldsymbol{W}, \boldsymbol{K}$ to $\boldsymbol{V}$, we first embed $\boldsymbol{W}_i, \boldsymbol{k}_i$ into a 32-dimensional space using one layer of $\mathrm{relu} \circ \mathrm{linear}(\cdot)$. These two 32-dimensional vectors are then concatenated into a 64-dimensional vector following by a batch-norm layer. Denote these 64-dimensional embedding for each unit as $\boldsymbol{E} = [\boldsymbol{e}_1, ..., \boldsymbol{e}_N]^\top$. We encode them into $N \times 1$-dimensional outputs $\boldsymbol{O}$ using a scaled product attention with value, key, query being linear transformations of $\boldsymbol{E}$. Notice that we read off the balancing weights via $\boldsymbol{V}/h(\boldsymbol{X})\boldsymbol{W}$ and $h(\boldsymbol{X}) \succ \boldsymbol{0}$. As the optimal weights $\boldsymbol{\alpha}^* \succeq \boldsymbol{0}$, the values $\boldsymbol{V}$ should have the same sign as $\boldsymbol{W}$ in an element-wise fashion. Therefore to enforce this, we include another multiplier layer to obtain $\boldsymbol{V}$ from the outputs $\boldsymbol{O}$, namely, $\boldsymbol{V} = \mathrm{relu}(\boldsymbol{O}\boldsymbol{W})$.

Normalization. As the optimal balancing weights is in $\mathbb{A} = \{\boldsymbol{0} \preceq \boldsymbol{\alpha} \preceq \boldsymbol{1},\ \sum_{i\in\mathbb{T}} \alpha_i = \sum_{i\in\mathbb{C}} \alpha_i = 1\}$, we normalize the read-off balancing weights during inference. In particular, in Algorithm 1 and Algorithm 3, after setting $\boldsymbol{\alpha}^* = \lambda \cdot \boldsymbol{V}/h(\boldsymbol{X})\boldsymbol{W}$, we project it into $\mathbb{A}$ by taking $\max(\boldsymbol{\alpha}^*, \boldsymbol{0})$ and normalizing the treated and control group to sum up to 1.

Hyper-parameters. For both Algorithm 1 and Algorithm 2, we search for the optimal penalty $\lambda > 0$ from range $[\lambda_{\min}, \lambda_{\max}]$ by exponentially increasing it from $\lambda_{\min}$ to $\lambda_{\max}$. On the same dataset, this range remains the same for both algorithms (and all variations, if applicable). The following table summarizes the values of $\lambda_{\min}$ to $\lambda_{\max}$ for different datasets.

Training and Evaluations. For all the experiments, we use a cosine annealing schedule for the learning rate from $l_{\max}$ to $l_{\min}$ during the first half of the training epochs. Then the learning rate is fixed to $l_{\min}$ for the second half of the training epochs. The exact values of $l_{\max}$ and $l_{\min}$ for

[3] In datasets with large sample sizes, techniques for efficient transformers (Child et al., 2019; Kitaev et al., 2020; Katharopoulos et al., 2020; Sun et al., 2023) can be applied to accelerate our method.

Dataset	$\lambda_{\min}$	$\lambda_{\max}$
Simulation A	1e-6	1e-2
Simulation B	1e-6	1e-2
IHDP	1	1000
IHDP-resmapled	1e-5	1000
Twins	1e-8	1e-2
LaLonde CPS	1e-10	5e-6
LaLonde PSID	1e-10	5e-6
ACIC	1e-6	100

Table 1: Search range for λ in different datasets.

different datasets can be found in the codebase. For Algorithm 1, we train for $20,000$ epochs on all datasets. For Algorithm 2, we train for $4,000$ epochs on all datasets.

For evaluating the results of Algorithm 2, we choose the best hyper-parameters based on the mean absolute error on the validation sets of datasets and report the results on the testing sets of datasets. For evaluating the results of Algorithm 1, if the setting contains multiple datasets (Simulation A, Simulation B, IHDP-resampled, ACIC), we choose the best hyper-parameters based on the mean absolute error on the validation sets of datasets and report the results on the testing sets of datasets. Note that even though IHDP contains multiple datasets, they all share the same sets of covariates and treatments. Therefore we treat it the same as settings with one dataset for Algorithm 1. On these datasets (IHDP, Twins, LaLonde CPS, LaLonde PSID), we choose the best hyper-parameters based on the reported results.

H.2 Baselines

IPW and Self-Normalized IPW. For both IPW and self-normalized IPW, we first standardized the covariates $\boldsymbol{X}$. Then we fit a random forest classifier on the data to predict propensity scores. The depth of the random forest classifier is chosen in the same way as the hyper-parameter λ is chosen in CInA, which we described above.

DML. For DML, we use the implementation of Battocchi et al. (2019). In particular, we consider three models: `LinearDML`, `CausalForestDML`, `KernelDML`. Similar as above, when a validation set of datasets is present, we report the results based on the best of these three models in terms of validation MAE. Otherwise we report based on the best performance on the reported dataset. However, in simulation A, we only use `LinearDML` as the outcome model is linear.

SVM. For this baseline, we first standardized the covariates $\boldsymbol{X}$. Then we solve the dual SVM problem in Eq. (2), where the kernel is defined using ϕ given in Eq. (9) on the standardized data. We use the support vector classifier (Pedregosa et al., 2011) with a precomputed kernel. The maximum number of iterations is capped with a hard limit of $50,000$. The reported results are based on λ choosen in the same way as CInA described above.

H.3 Dataset Augmentation

In our experiments in Section 5.1 and certain datasets in Section 5.3 using the multi-dataset version of CInA, we implemented a new type of data augmentation. As we observe that the network can learn how to balance on a set of datasets using very few training steps, we propose to reshuffle amongst different datasets in every epoch. This essentially creates a "new" set of datasets by combining units from different datasets. Intuitively, this augments the number of covariate balancing problems that the model has to learn to solve without actually needing to acquire more data. However, we note that this technique is only applied if different datasets from the same experiment share the same causal graph. If different datasets contain very different causal structures such as **ER-5000** in Section 5.2 and **ACIC** in Section 5.3, this shuffling is not used as it would create covariate balancing problem that does not aid learning. The main intuition is that if we reshuffle units among these datasets, units in a reshuffled dataset could follow different causal graphs, which means there is potentially no underlying causal structure that can explain the data.

I ADDITIONAL EMPIRICAL RESULTS

I.1 COMPARISON TO DRAGONNET AND RIESZNET

Method	Simulation-A	ER-5000	IHDP
Naive	0.172 ± 0.03	50.27 ± 5.97	0.259 ± 0.01
IPW	0.304 ± 0.03	27.42 ± 3.19	0.766 ± 0.02
Self-normalized IPW	0.158 ± 0.03	49.99 ± 5.88	0.141 ± 0.00
DML	0.094 ± 0.01	11.13 ± 3.17	0.585 ± 0.03
DragonNet	0.386 ± 0.01	11.21 ± 3.17	0.146 ± 0.01
RieszNet	**0.045 ± 0.01**	12.90 ± 4.54	**0.110 ± 0.01**
SVM	0.015 ± 0.00	11.09 ± 3.13	1.202 ± 0.05
Ours	0.126 ± 0.02	N/A	**0.114 ± 0.01**
Ours (ZS)	0.147 ± 0.01	11.50 ± 1.85	N/A
Ours (ZS-S)	N/A	**2.66 ± 0.33**	N/A
Mean	N/A	17.88 ± 1.83	N/A

Table 2: ATE MAE comparison of different methods on the "Simulation-A", "ER-5000", and "IHDP" datasets.

In this section, we further compare two additional baselines, DragonNet (Shi et al., 2019) and RieszNet (Chernozhukov et al., 2022), both of which were considered strong neural estimation methods for per-dataset causal inference. Results for **IHDP** dataset were directly cited from (Shi et al., 2019; Chernozhukov et al., 2022), following their best performing models. Furthermore, we also compare to **Simulation-A-Multi+OOD+diff_size**, and **ER-5000**, both are the most general synthetic settings in Section 5. On **Simulation-A-Multi+OOD+diff_size**, *CINA (ZS)* outperforms *DragonNet*, while *RieszNet* outperforms both *DragonNet* and *CINA (ZS)* method. On both **ER-5000** and **IHDP**, *CINA (ZS)* is on par with or outperforms *DragonNet* and *RieszNet*, while *CINA (ZS-S)* massively outperforms the other methods on **ER-5000**.

I.2 LARGER SCALE EXPERIMENTS ON 10K ACIC 2018, WITH CROSS-DATASET GENERALIZATION

Method	ATE MAE	Inference time on new data (s)	Pretraining time (s)
Naive	13.07 ± 8.25	0.005	N/A
IPW	10.29 ± 5.94	48.927	N/A
Self-normalized IPW	10.30 ± 5.90	49.322	N/A
DML	8.572 ± 8.96	7391.743	N/A
RieszNet	69.39 ± 31.9	8157.498	N/A
Ours (ZS)	**1.460 ± 0.48**	78.503	1800
Ours (ZS-S)	**1.361 ± 0.42**	77.546	1800
Ours (ZS-ER)	1.718 ± 0.74	78.085	1800
Ours (ZS-S-ER)	1.702 ± 0.74	77.947	1800

Table 3: Comparison of different methods on the 10k ACIC 2018 dataset.

To demonstrate the performance of our method on larger version of **ACIC 2018**, we produce additional experiment using the 10k-size datasets of ACIC (Shimoni et al., 2018), which is a commonly used scale considered in the literature (Shi et al., 2019; Mahajan et al., 2022). Note that instead of only selecting a subset of datasets in **ACIC 2018** as in (Shi et al., 2019; Mahajan et al., 2022), we make use of all datasets of size 10k generated by (Shimoni et al., 2018) that has polynomial link functions as training datasets, and all datasets of size 10k with exponential link functions as test datasets.

In this setting, we also compare two new variants of our method, *CINA (ZS-ER)* and *CINA (ZS-S-ER)*, that are fully trained on a larger-scale, 200-dimensional ER-5000 dataset Section 5.2 under both unsupervised and supervised settings, respectively. After pre-training, *CINA (ZS-ER)* and

CINA (ZS-S-ER) are applied directly to all ACIC 2018 test sets. This will help us to demonstrate whether the model can show generalization ability across datasets. All CINA-related methods are trained for a fixed time budget (1800 seconds), which is significantly shorter than the full training time of DML and RieszNet. As shown in Table 2, both *CINA (ZS)* and *CINA (ZS-S)* significantly outperforms all baselines. The *CINA (ZS-ER)* and *CINA (ZS-S-ER)* methods give marginally worse performance than *CINA (ZS)* and *CINA (ZS-S)*, but still out-performs the other baselines by a clear margin.