IDENTIFYING TREATMENT EFFECTS UNDER UNOBSERVED CONFOUNDING BY CAUSAL REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

As an important problem of causal inference, we discuss the estimation of treatment effects under the existence of unobserved confounding. By representing the confounder as a latent variable, we propose Counterfactual VAE, a new variant of variational autoencoder, based on recent advances in identifiability of representation learning. Combining the identifiability and classical identification results of causal inference, under mild assumptions on the generative model and with small noise on the outcome, we theoretically show that the confounder is identifiable up to an affine transformation and then the treatment effects can be identified. Experiments on synthetic and semi-synthetic datasets demonstrate that our method matches the state-of-the-art, even under settings violating our formal assumptions.

1 INTRODUCTION

Causal inference (Imbens & Rubin [2015], Pearl [2009]), i.e., estimating causal effects of interventions, is a fundamental problem across many domains. In this work, we focus on the estimation of treatment effects, e.g., effects of public policies or a new drug, based on a set of observations consisting of binary labels for treatment / control (non-treated), outcome, and other covariates. The fundamental difficulty of causal inference is that we never have observations of counterfactual outcomes, which would have been if we had made another decision (treatment or control). While the ideal protocol for causal inference is randomized controlled trials (RCTs), they often have ethical and practical issues, or are prohibitively expensive. Thus, causal inference from observational data is indispensable, though they introduce other challenges. Perhaps the most crucial one is confounding: there might be variables (called confounders) that causally affect both the treatment and the outcome, and spurious correlation follows.

Most of works in causal inference rely on the unconfoundedness assumption that appropriate covariates are collected so that the confounding can be controlled by conditioning on or adjusting for those variables. This is still challenging, due to systematic difference of the distributions of the covariates between the treatment and control groups. One classical way of dealing with this difference is re-weighting (Horvitz & Thompson [1952]). There are semi-parametric methods, which have better finite sample performance, e.g. TMLE (Van der Laan & Rose [2011]), and also non-parametric, tree-based methods, e.g. Causal Forests (CF) (Wager & Athey [2018]). Notably, there is a recent rise of interest in representation learning for causal inference starting from Johansson et al. [2016].

There are a few lines of works that challenge the difficult but important problem of causal inference under unobserved confounding. Without covariates we can adjust for, many of them assume special structures among the variables, such as instrumental variables (IVs) (Angrist et al. [1996]), proxy variables (Miao et al. [2018]), network structure (Ogburn [2018]), and multiple causes (Wang & Blei [2019]). Among them, instrumental variables and proxy (or surrogate) variables are most commonly exploited. Instrumental variables are not affected by unobserved confounders, influencing the outcome only through the treatment. On the other hand, proxy variables are causally connected to unobserved confounders, but are not confounding the treatment and outcome by themselves. Other methods use restrictive parametric models (Allman et al. [2009]), or only give interval estimation (Manski [2009], Kallus et al. [2019]).
In this work, we address the problem of estimating treatment effects under unobserved confounding. We further discuss the individual-level treatment effect, which measures the treatment effect conditioned on the covariate, for example, on a patient’s personal data. To model the problem, we regard the covariate as a proxy variable and the confounder as a latent variable in representation learning.

Our method particularly exploits the recent advance of identifiability of representation learning for VAE (Khemakhem et al., 2020). The hallmark of deep neural networks (NNs) might be that they can learn representations of data. It is desirable that the learned representations are interpretable, that is, in approximately the same relationship to the latent sources for each down-stream task. A principled approach to this is identifiability, that is, when optimizing our learning objective w.r.t. the representation function, only a unique optimum will be returned. Our method builds on this and further provides the stronger identifiability of representations that is needed in causal inference.

The proposed method is also based firmly on the well-established results in causal inference. In many works exploiting proxies, it is assumed that the proxies are independent of the outcome given the confounder (Greenland, 1980; Rothman et al., 2008; Kuroki & Pearl, 2014). This also motivates our method. Further, our method naturally combines a new VAE architecture with the classical result of Rosenbaum & Rubin (1983) regarding the sufficient information for identification of treatment effects, showing identifiability proof of both latent representations and treatment effects.

The main contributions of this paper are as follows: 1) interpretable, causal representation learning by a new VAE architecture for estimating treatment effects under unobserved confounding; 2) theoretical analysis of the identifiability of representation and treatment effect; 3) experimental study on diverse settings showing performance of state-of-the-art.

2 RELATED WORK

Identifiability of representation learning. With recent advances in nonlinear ICA, identifiability of representations is proved under a number of settings, e.g., auxiliary task for representation learning (Hyvärinen & Morioka, 2016; Hyvärinen et al., 2019) and VAE (Khemakhem et al., 2020). Recently, Roeder et al. (2020) extends the result to include a wide class of state-of-the-art deep discriminative models. The results are exploited in bivariate causal discovery (Wu & Fukumizu, 2020) and structure learning (Yang et al., 2020). To the best of our knowledge, this work is the first to explore this new possibility in causal inference.

Representation learning for causal inference. Recently, researchers start to design representation learning methods for causal inference, but mostly limited to unconfounded settings. Some methods focus on learning a balanced covariate representation, e.g., BLR/BNN (Johansson et al., 2016), and TARnet/CFR (Shalit et al., 2017). Adding to this, Yao et al. (2018) also exploits the local similarity of between data points. Shi et al. (2019) uses similar architecture to TARnet, considering the importance of treatment probability. There are also methods using GAN (Yoon et al., 2018, GANITE) and Gaussian process (Alaa & van der Schaar, 2017). Our method adds to these by also tackling the harder problem of unobserved confounding.

Causal inference with auxiliary structures. Both our method and CEVAE (Louizos et al., 2017) are motivated by exploiting proxies and use VAE as a learning method. However, CEVAE assumes a specific causal graph where the covariates should be independent of the treatment given the confounder. Further, CEVAE relies on the assumption that VAE can recover the true latent distribution. Kallus et al. (2018) uses matrix factorization to infer the confounders from proxy variables, and gives consistent ATE estimator and its error bound. Miao et al. (2018) established conditions for identification using more general proxies, but without practical estimation method. Note that, two active lines of works in machine learning exist in their own right, exploiting IV (Hartford et al., 2017) and network structure (Veitch et al., 2019).

3 SETUP AND PRELIMINARIES

3.1 TREATMENT EFFECTS AND CONFOUNDERS

Following Imbens & Rubin (2015), we begin by introducing potential outcomes (or counterfactual outcomes) \( y(t), t = 0, 1 \). \( y(t) \) is the outcome we would observe, if we applied treatment value \( t \).
Note that, for a unit under research, we can observe only one of \( y(0) \) or \( y(1) \), corresponding to which factual treatment we have applied. This is the fundamental problem of causal inference.

We write expected potential outcomes, conditioned on covariate(s) \( x \) as \( \mu_t(x) = E(y(t)|x = x) \). The estimands in this work are the causal effects, which are Conditional Average Treatment Effect (CATE) and Average Treatment Effect (ATE) defined by
\[
\tau(x) = \mu_1(x) - \mu_0(x), \quad ATE = E(\tau(x))
\]  
(1)

CATE can be understood as an individual-level treatment effect, if conditioned on high dimensional and highly diverse covariates.

In general, we need three assumptions for identification [Rubin 2005]. There should exist variable \( z \in \mathbb{R}^n \) satisfies ignorability \( (y(0), y(1)) \perp \!\!\!\!\perp t | z \) and positivity \( (\forall z, t : p(t = t | z = z) > 0) \), and also given the consistency of counterfactuals \( (y = y(t) \text{ if } t = t) \) (See Appendix for explanations). Then, treatment effects can be identified by:
\[
\mu_t(x) = E(E(y(t)|x, x = x)) = E(E(y|z, x = x, t = t)) = \int \int p(y|z, x, t)dy dz \big| p(z|x)
\]
(2)
The second equality uses the three conditions. We say that strong ignorability holds when we have both ignorability and positivity. In this work, we consider unobserved confounding, that is, we assume the existence of confounder(s) \( z \), satisifying the three conditions, but it is (partially) unobserved.

The following theorem adapted from Rosenbaum & Rubin (1983) is central to causal inference and we will use it for motivating and justifying our method. Such function \( b(z) \) is called a balancing score (of \( z \)). Obviously, the propensity score \( e(z) := p(t = 1|z) \), the propensity of assigning the treatment given \( z \), is a balancing score (with \( f \) be the identity function).

**Theorem 1** (Balancing score). Let \( b(z) \) be a function of random variable \( z \). Then \( t \perp z | b(z) \) if and only if \( f(b(z)) = p(t = 1|z) := e(z) \) for some function \( f \) (or more formally, \( e(z) \) is \( b(z) \)-measurable). Assume further that \( z \) satisfies strong ignorability, then so does \( b(z) \).

### 3.2 Variational autoencoders

Variational autoencoders (VAEs) [Kingma et al. 2019] are a class of latent variable models with latent variable \( z \), and observed variable \( y \) is generated by the decoder \( p_{\theta}(y|z) \). The variational lower bound of the log-likelihood is written as:
\[
\log p(y) \geq \log p(y) - D_{KL}(q_{\phi}(z|y)||p(z|y)) = \mathbb{E}_{z \sim q} \log p_{\theta}(y|z) - D_{KL}(q_{\phi}(z|y)||p(z)),
\]
(3)
where the encoder \( q_{\phi}(z|y) \) is introduced to approximate the true posterior \( p(z|y) \) and \( D_{KL} \) denotes KL divergence. The decoder \( p_{\theta} \) and encoder \( q_{\phi} \) are usually parametrized by NNs. We will omit the parameters \( \theta, \phi \) in notations when appropriate. Using the reparameterization trick [Kingma & Welling 2014] and optimizing the evidence lower bound (ELBO) \( \mathbb{E}_{y \sim D} (L(y)) \) with data \( D \), we train the VAE efficiently. Conditional VAE (CVAE) adds a conditioning variable \( c \) to \( (3) \) (See Appendix for details).

As mentioned, identifiable VAE (iVAE) [Khemakhem et al. 2020] provides the first identifiability result for VAE, using auxiliary variable \( u \). It assumes \( y \perp z | u \), that is, \( p(y|z, u) = p(y|z) \). The variational lower bound is
\[
\log p(y|u) \overset{\text{var}}{\geq} \mathbb{E}_{z \sim q} \log p_f(y|z) - D_{KL}(q_{\phi}(z|y, u)||p_{T,\lambda}(z|u))
\]
(4)
where \( y = f(z) + \epsilon, \epsilon \) is additive noise and \( z \) has exponential family distribution with sufficient statistics \( T \) and parameter \( \lambda(u) \). Note that, unlike CVAE, the decoder does not depend on \( u \) due to the independence assumption.

Here identifiability means that the functional parameters \( (f, T, \lambda) \) can be identified (learned) up to a simple transformation.

### 4 VAE architecture for causal representation learning

---

1This allows the existence of observed confounders in \( x \). As we will see, since \( z \) is the latent variable(s) for VAE and is learned from covariates \( x \) by the VAE, it can contain all confounders in principle. Our method will extract the confounding part of \( x \) into \( z \).
The probabilistic model of our VAE follows naturally from the right most side of (2), where the involved distribution is \( p(y, z|x, t) = p(y|z, x, t)p(z|x, t) \). If we treat covariate \( x \) as a proxy and assume further the conditional independence \( y \perp x|z, t \) as in most work exploiting proxies, we have
\[
p(y, z|x, t) = p(y|z, t)p(z|x, t) \tag{5}
\]
A natural next step is to design a VAE for this joint distribution, which learns to recover a causal representation of \( z \). By “causal”, we mean the representation can be used to identify or estimate treatment effects. Recovering the true confounder \( z \) would be great, but this is not required, as shown in Theorem 1, which says, for identification of treatment effects, we only need to have \( b(z) \), a causal representation of \( z \), which contains the information of propensity score \( e(z) \), the part of \( z \) that is relevant to treatment assignment.

We are a step away from the VAE architecture now. Note that [5] has similar factorization with iVAE: \( p(y, z|u) = p(y|z)p(z|u) \) from \( y \perp u|z \), meaning we can use our covariate \( x \) as auxiliary variable \( u \) in iVAE from the independence assumption of proxy variable. Further from the conditioning on \( t \) in (5), we design a VAE architecture as a combination of CVAE and iVAE, with treatment \( t \) and covariate \( x \) as conditioning and auxiliary variable, respectively. The ELBO can be derived as
\[
\log p(y|x, t) \geq \log p(y|x, t) - D_{KL}(q(z|x, y, t)||p(z|x, y, t)) = \mathbb{E}_{z \sim q} \log p(y|z, t) - D_{KL}(q(z|x, y, t)||p(z|x, t)) := L_{CFVAE}(x, y, t). \tag{6}
\]
As in iVAE, the decoder drops the dependence on \( x \). We name this architecture the Counterfactual VAE (CFVAE). Figure 1 depicts the relationship of CVAE, iVAE, and CFVAE.

We detail parameterization of CFVAE. The decoder \( p_{\theta,\phi}(y|z, t) \), conditional prior \( p_{h, k}(z|x, t) \), and encoder \( q_{\gamma, \phi}(z|x, y, t) \) are factorized Gaussians, i.e., a product of 1-dimensional Gaussian distributions. This is not restrictive if the mean and variance are given by arbitrary nonlinear functions.
\[
y|z, t \sim \prod_{j=1}^{d} N(y_j; f_j, g_j), \quad z|x, t \sim \prod_{i=1}^{n} N(z_i; h_i, k_i), \quad z|x, y, t \sim \prod_{i=1}^{n} N(z_i; r_i, s_i). \tag{7}
\]
\( \theta = (f, g, h, k) \) and \( \phi = (r, s) \) are functional parameters given by NNs which take the respective conditional variables as inputs (e.g. \( h := (h_i(x, t))^T \)).

5 IDENTIFYING REPRESENTATION AND TREATMENT EFFECTS

In the following, we will show that CFVAE can identify the latent variable up to an affine transformation (Sec. 5.1), it can learn a balancing score as a causal representation, and its decoder is a valid estimator for potential outcomes (Sec. 5.2).

5.1 IDENTIFIABILITY OF REPRESENTATION

In this subsection, we show that CFVAE can identify latent variable \( z \) up to an element-wise affine transformation when the noise on the outcome is small. Based on this result, we can gain insight on how to make CFVAE learn a balancing score.

Our starting point is the following theorem showing the identifiability of our learning model, adapted from Theorem 1 in [Khemakhem et al. 2020], by adding conditioning on \( t \).

**Theorem 2.** Given the family \( p_{\theta}(y, z|x, t) \) specified by (5) and (7) for \( t = 0, 1 \), assume
1) \( f_j(z) := (f_j(z, t))^T \) is injective;
2) \( g_j(z) = \sigma_{y, t} \) is constant (i.e. \( g_j(z, t) = \sigma_{y, t} \));
3) \( \lambda_i(x) := (h(x, t), k(x, t))^T \), which is seen as a random variable, is not degenerate.

We specified factorized Gaussians in (7) and they show good performance in our experiments. But Corollary 1 and Theorem 2 can be extended to more general exponential families, see [Khemakhem et al. 2020].
Then, given \( t = t \), the family is identifiable up to an equivalence class. That is, for \( t = 0,1 \), if \( p\theta_t(\mathbf{y}|\mathbf{x},t=0) = p\theta_t(\mathbf{y}|\mathbf{x},t=1) \), we have the relation between parameters
\[
f_t^{-1}(y_t) = \mathbf{A}_t f_t^{-1}(y_t) + \mathbf{b}_t := \mathbf{A}_t(f_t^{-1}(y_t))
\]
where \( p(y_t) := p(y|t) \). \( \mathbf{A}_t \) is an invertible \( n \)-square matrix and \( \mathbf{b}_t \) is a \( n \)-vector.

Similarly to Sorrenson et al. [2019], we can further show that \( \mathbf{A}_t = \text{diag}(\mathbf{a}_t) \) is a diagonal matrix. By a slight abuse of symbol, we will overload \( | \) to make a shorthand for equations like (5), e.g., (8) can be written as \( f^{-1}(y) = \mathcal{A}(f^{-1}(y))|t \). Note that, by definition of inverse, we also have \( f = f \circ \mathcal{A}|t \).

The importance of model identifiability can be seen more clearly in the limit of small noise on \( y \). Corollary [1] can be easily understood by noting that after learning with small noise on \( y \), the encoder and decoder both degenerate to deterministic functions: in (7) \( g' = s' = 0 \), and \( \forall \mathbf{x}, \mathbf{z}'_t = f_t^{-1}(y) \). Note that, we only assume the VAE learns observational distributions \( p\theta_t(\mathbf{y}|\mathbf{x},t=t) \) the same as the truth, but this leaves room for latent distributions different to the truth.

**Corollary 1** (Identifiability of representation). For \( t = 0,1 \), assume 1) \( \sigma_{y,t} \rightarrow 0 \) and 2) CFV AE can learn a distribution \( p\theta_t = p\theta_0 \), then the latent variable \( \mathbf{z} \) and the mean parameter \( f_t \) of \( y \) can be identified up to an element-wise affine transformation: \( \mathbf{z} = \mathcal{A}(\mathbf{z}')|t \) and \( f_t = f \circ \mathcal{A}|t \).

This is a strong result for learning interpretable representation, but it is not enough for causal inference. To see this in a principled way, we recall the concept of balancing score. The recovered latent \( \mathbf{z}' \) in Corollary [1] is not a balancing score, due to the different \( \mathcal{A}_t \) for \( t = 0,1 \). If \( \mathbf{z}' \) were a balancing score, we would have \( \mathbb{I}\mathbb{I}\mathbf{z} = \mathbf{z}' \). However, given \( \mathbf{z}' = \mathbf{z}' \), \( \mathbf{z} = \text{diag}(\mathbf{a}_t)\mathbf{z}'' + \mathbf{b}_t \) is a deterministic function of \( t \), contradicting with \( \mathbb{I}\mathbb{I}\mathbf{z} = \mathbf{z}' \). (A more concrete analysis can be found in Appendix.)

This example also suggests that we will have a balancing score if we get rid of the dependence on \( t = t \). The next subsection discusses some assumptions on \( \mathbf{x} \) to remove the “\( |t \)” in Corollary 1.

### 5.2 Identification of Treatment Effects

The following definition will be used in Theorem 3. The importance of this definition is immediate from Theorem 1 that is, if a balancing covariate is also a function of \( \mathbf{z} \), then it is a balancing score.

**Definition 1** (Balancing covariate). Random variable \( \mathbf{x} \) is a balancing covariate of random variable \( \mathbf{z} \) if \( \mathbb{I}\mathbb{I}\mathbf{z} = \mathbf{x} \). We also simply say \( \mathbf{x} \) is balancing (or non-balancing if it does not satisfy this definition).

Given that a balancing score of the true confounder is sufficient for strong ignorability, a natural and interesting question is that, does a balancing covariate of the true confounder also satisfies strong ignorability? The answer is no. To see why, and also to better understand the significance of Theorem 3 we give Proposition 1 indicating that a balancing covariate of the true confounder might not even satisfy ignorability. We also refer readers to Appendix 8.5 where we examine two important special cases of balancing covariate, one of those is noiseless proxy, which might not satisfy positivity.

**Proposition 1.** Let \( \mathbf{x} \) be a balancing covariate of \( \mathbf{z} \). If \( \mathbf{z} \) satisfies ignorability and \( y(0),y(1) \mathbb{I}\mathbb{I}\mathbf{z} = \mathbf{z}, t \) then \( \mathbf{x} \) satisfies ignorability.

Given this proposition, we know our assumptions are weaker than ignorability, and our method can work under unobserved confounding (\( \mathbf{x} \) might not satisfy ignorability). Note the independence \( y(0),y(1) \mathbb{I}\mathbf{z} = \mathbf{x}, t \) in this proposition implies \( y \mathbb{I}\mathbf{z} = \mathbf{x}, t \) assumed by CFV AE. From the decompositions rule of conditional independence, we have \( y(0),y(1) \mathbb{I}\mathbf{z} = \mathbf{x}, t \) \( \implies \forall t,y(t) \mathbb{I}\mathbf{z} = \mathbf{x}, t \) \( \implies \forall t,y(t) \mathbb{I}\mathbf{z} = \mathbf{x}, t \). Using only two of them, and with the consistency of counterfactual, we have \( \forall t,y(t) \mathbb{I}\mathbf{z} = \mathbf{x}, t \) \( \implies \forall t,y(t) \mathbb{I}\mathbf{z} = \mathbf{x}, t \) \( \implies y \mathbb{I}\mathbf{z} = \mathbf{x}, t \). In general, the other two independence \( y(0) \mathbb{I}\mathbf{z} = \mathbf{x}, t = 1 \) and \( y(1) \mathbb{I}\mathbf{z} = \mathbf{x}, t = 0 \) do not hold, and thus \( y \mathbb{I}\mathbf{z} = \mathbf{x}, t \) \( \implies y(0),y(1) \mathbb{I}\mathbf{z} = \mathbf{x}, t \).

As shown in the following theorem, for any balancing covariates, CFV AE can identify treatment effects, by learning a balancing score of the true confounder as latent representation. Note that

\[ \mathbf{b}_t \] is another set of parameters giving the same distribution, which is learned by VAE. In this paper, symbol “\( \prime \)” (prime) always indicates parameters (variables, etc.) learned/recovered by VAE.
assumption 3) is nontrivial, it requires that the true data generating distribution is as following. And thus identification of treatment effects follows from the identifiability of our model.

**Theorem 3 (Identification with balancing covariate).** Assume

1) the same as Theorem 2 and Corollary 7

2) $h$, $k$ in (7) depend on $x$ but not $t$ (i.e. $h_i(x, t) = h_i(x)$ and same for $k$);

3) The data generating process can be parametrized by family $p_0(y, z|x, t)$ specified above;

4) $z$ satisfies strong ignorability and $x$ is a balancing covariate of $z$.

Then, for both $t = 0, 1$, we have $z = \text{diag}(a)y' + b := A(z')$, and $z'$ satisfies strong ignorability. We identify the potential outcomes by $\mu_t(x) = \mathbb{E}(E(y'|x, t = \hat{t}))) = \mathbb{E}(f_i^t(r_i^t(x, y_t))|x = x)$.

The result may be more easily understood as following. Now with the same $A$ for both treatment groups $t$, given observation $y_t$, the counterfactual prediction given by CFVAE is the same as truth: $y_{1-\hat{t}} = f_{1-\hat{t}}(z_{1-\hat{t}}) = f_{1-\hat{t}} \circ A^{-1}(z_{1-\hat{t}}) = y(1 - t)$ (Also compare this to Appendix 8.4). We can identify potential outcomes, using $r_i^t(x, y_t) = z_t'$ in the last equality, by

$$\mu_t(x) = \mathbb{E}(y_t|x = x) = \mathbb{E}(y_t|x = x) = \mathbb{E}(f_i^t(r_i^t(x, y_t))|x = x)$$ (9)

Note that counterfactual assignment $t = \hat{t}$ may or may not be the same as factual $t$. The algorithm for estimation CATE and ATE is as following. After training CFVAE, we feed data $D = \{(x, y_t) := (x, y, t)\}$ into the encoder, and draw sample from it: $q(z'|x = x, y = y, t = \hat{t}) = \delta(z' - r_i^t(x, y_t))$ ($\delta$ denotes delta function). Then, setting $t = \hat{t} \in \{0, 1\}$ in the decoder, feed posterior sample $z_t'$ in (7), we get counterfactual prediction $p(y'|z' = z'_t, t = \hat{t}) = \delta(y'_t - f_i^t(z'_t))$. Finally, we estimate ATE by taking average $\mathbb{E}_D(y'_1 - y'_0)$, and CATE by $\mathbb{E}_{\{D|x=x\}}(y'_1 - y'_0)$, adding conditioning on $x$.

A caveat is that (9) requires post-treatment observation $y_t$. Often, it is desirable that we can also have pre-treatment prediction for a new subject, with only the observation of its covariate $x = x$. To this end, we use conditional prior $p(z'|x)$ as a pre-treatment predictor for $z'$: input $x$ and draw sample from $p(z'|x = x)$ instead of $q$, and all the others remain the same. Since the ELBO has a KL term between $p(z'|x)$ and $q$, the two distributions should not be very different, and we will also have sensible pre-treatment estimation of treatment effects.

Although our method works under unobserved confounding, it still formally requires small outcome noise and balancing covariate. However, experiments show our method can work very well with large outcome noise, and the covariates can be non-balancing and also directly affect the outcome, including general proxies, IVs, and even networked data.

### 6 Experiments

As in previous works (Shalit et al. 2017, Louizos et al. 2017), we report the absolute error of ATE $\epsilon_{ATE} := |\mathbb{E}_D(y(1) - \overline{y}(0)) - \mathbb{E}_D(y'_1 - y'_0)|$, and the square root of empirical PEHE (Hill 2011) $\epsilon_{PEHE} := \mathbb{E}_D((y(1) - \overline{y}(0)) - (y'_1 - y'_0))^2$ for individual-level treatment effects.

Unless otherwise indicated, for each function $f, g, h, k, r, s$ in (7), we use a multilayer perceptron (MLP) that has 3*200 hidden units with ReLU activation, and $h, k$ depend only on $x$. The Adam optimizer with initial learning rate $10^{-4}$ and batch size 100 is employed. More details on hyper-parameters and experimental settings are given in each experiment, and are explained in Appendix.

All experiments use early-stopping of training by evaluating the ELBO on a validation set. We evaluate the post-treatment performance on training and validation set jointly (This is non-trivial. Recall the fundamental problem of causal inference). The treatment and (factual) outcome should not be observed for pre-treatment predictions, so we report them on a testing set.

#### 6.1 Synthetic dataset

We generate data following (10) with $z$, $y$ 1-dimensional and $x$ 3-dimensional. $\mu_t$ and $\sigma_t$ are randomly generated in range $(-0.2, 0.2)$ and $(0, 0.2)$, respectively. The functions $h, k, l, s$ are linear with random coefficients. The outcome model is built for the two treatments separately, i.e.
\( f(z, t) := f_t(z), t = 0, 1. \) We generate two kinds of outcome models, depending on the type of \( f_t: \) linear and nonlinear outcome models use random linear functions and NNs with random weights, respectively.

\[
\begin{align*}
  x &\sim \prod_{i=1}^{3} \mathcal{N}(\mu_i, \sigma_i); \quad z|x \sim \mathcal{N}(h(x), \beta(k(x))); \\
  t|x, z &\sim \text{Bern}(\text{Logistic}(l(x, z))); \quad y|z, t \sim \mathcal{N}(C^{-1}_t f(z, t), \alpha).
\end{align*}
\]  

(10)

We adjust the outcome and proxy noise level by \( \alpha, \beta \) respectively. The output of \( f_t \) is normalized by \( C_t := \text{Var}_z \{ f_t(z) \} \). This means we need to use \( 0 \leq \alpha < 1 \) to have a reasonable level of noise on \( y \) (the scales of mean and variance are comparable). Similar reasoning applies to \( z|x \); outputs of \( h, k \) have approximately the same range of values since the functions’ coefficients are generated by the same weight initializer.

We experiment on three different causal settings (indicated in Italic). To introduce \( x \) as \( IV \), we generate another 1-dimensional random source \( w \) in the same way as \( x \), and use \( w \) instead of \( x \) to generate \( z|w \sim \mathcal{N}(h(w), \beta(k(w))) \). Besides taking inputs \( x, z \) in \( l \), we consider two special cases: \( l := l(x) \) (fully satisfies ignorability) and \( l := l(z) \) (unobserved confounder \( z \) and non-balancing proxy \( x \) of \( z \)). Except indicated above, other aspects of the models are specified by (10). See Appendix for graphical models of these three cases.

In each causal setting, and with the same kind of outcome models and noise levels \( (\alpha, \beta) \), we evaluate CFVAE and CEVAE on 100 random data generating models, with different sets of functions \( f, h, k, l \) in (10). For each model, we sample 1500 data points, and split them into 3 equal sets for training, validation, and testing. Both the methods use 1-dimensional latent variable in VAE. For fair comparison, all the hyper-parameters, including type and size of NNs, learning rate, and batch size, are the same for both the methods.

Figure 3 shows our method significantly outperforms CEV AE on all cases. Each method works the best under ignorability, as expected. The performances of our method on IV and proxy settings match that of CEV AE under ignorability, showing the effective deconfounding. Figure 2 shows our method learns highly interpretable representation as an approximate affine transformation of the true latent value. To our surprise, CEV AE is also possible to achieve this when both noises are small, though the quality of recovery is lower than CFV AE. The relationship to the true latent is significantly obscured under IVs, because the true latent is correlated to IVs only given \( t \), while we model it by \( p(z'|x) \) as required by Theorem 3.

We can see our method and also CEV AE are very robust w.r.t. both outcome and proxy noise. This may due to the good probabilistic modeling of the noise by VAE. Still, we can see in Appendix that the noise level affects how well we recover the latent variable.

### 6.2 IHDP Benchmark Dataset

The IHDP dataset [Hill, 2011] is widely used to evaluate machine learning based causal inference methods, e.g. [Shalit et al., 2017]; [Shi et al., 2019]. Here, ignorability holds given the covariates. See Appendix for detailed descriptions. Note, however, that this dataset violates our assumption \( y \perp \!\!\!\perp x|z, t \), since the covariates \( x \) directly affect the outcome. To overcome this, we add two components introduced by [Shalit et al., 2017] into our method. First, we build two outcome functions \( f_t(z), t = 0, 1 \) in our learning model (7), using two separate NNs.
Second, we add to our ELBO (6) a regularization term, which is the Wasserstein distance (Cuturi, 2013) between the learned \( p(z'|x, t = 0) \) and \( p(z'|x, t = 1) \). We find higher than 1-dimensional latent variable in CFVAE gives better results, very possibly due to the mismatched latent distribution: the confounder race is discrete but we use Gaussian latent variable. We report results with 10-dimensional latent variable.

As shown in Table 1, the proposed CFVAE matches the state-of-the-art methods under model mis-specification. This robustness of VAE was also observed by Louizos et al. (2017), except GANITE (Yoon et al., 2018) and CEV AE (Louizos et al., 2017). Bold indicates method(s) that are significantly better than all the others. The results of the other methods are taken from Shalit et al. (2017), except GANITE (Yoon et al., 2018) and CEVAE (Louizos et al., 2017).

<table>
<thead>
<tr>
<th>Method</th>
<th>TMLE</th>
<th>BNN</th>
<th>CFR</th>
<th>CF</th>
<th>CEVAE</th>
<th>GANITE</th>
<th>Ours*</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \epsilon_{ATE} )</td>
<td>NA/30 ± 0.1</td>
<td>4.2 ± 0.05</td>
<td>0.37 ± 0.03</td>
<td>2.72 ± 0.01</td>
<td>25 ± 0.01</td>
<td>40 ± 0.01</td>
<td>18 ± 0.01</td>
</tr>
<tr>
<td>( \sqrt{\text{PEHE}} )</td>
<td>NA/50 ± 2</td>
<td>2.1 ± 1/2.2 ± 1</td>
<td>0.76 ± 0.05</td>
<td>0.71 ± 0.02</td>
<td>3.8 ± 0.35</td>
<td>2.6 ± 1/2.7 ± 1</td>
<td>2.4 ± 0.19/0.9 ± 0.4</td>
</tr>
</tbody>
</table>

### 6.3 Pokec Social Network Dataset

Pokec (Leskovec & Krevl, 2014) is a real world social network dataset. We experiment on a semi-synthetic dataset based on Pokec, which was introduced in Veitch et al. (2019), and use exactly the same pre-processing and generating procedure. The pre-processed network has about 79,000 vertices (users) connected by 1.3 × 10^6 undirected edges. The subset of users used here are restricted to three living districts that are within the same region. The network structure is expressed by binary adjacency matrix \( G \). Following Veitch et al. (2019), we split the users into 10 folds, test on each fold and report the mean and std of pre-treatment ATE predictions. We further separate the rest of users (in the other 9 folds) by 6 : 3, for training and validation. Table 2 shows the results. Our method is the best compared with the methods specialized for networked data. We report pre-treatment PEHE of our method in the Appendix, while Veitch et al. (2019) does not give individual-level prediction.

Table 2: Pre-treatment ATE on Pokec. Ground truth is 1. “Unadjusted” estimates ATE by \( \mathbb{E}_D(y_1) - \mathbb{E}_D(y_0) \). “Parametric” is a stochastic block model for networked data (Gopalan & Blei, 2013). “Embed-” denotes the best alternatives given by Veitch et al. (2019). Bold indicates method(s) that are significantly better than all the others. 20-dimensional latent variable in CFVAE works better, and its result is reported. The results of the other methods are taken from Veitch et al. (2019).

<table>
<thead>
<tr>
<th></th>
<th>Unadjusted</th>
<th>Parametric</th>
<th>Embed-Reg.</th>
<th>Embed-IPW</th>
<th>Ours*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>4.34 ± 0.05</td>
<td>4.06 ± 0.01</td>
<td>2.77 ± 0.35</td>
<td>3.12 ± 0.06</td>
<td>2.08 ± 0.32</td>
</tr>
<tr>
<td>District</td>
<td>4.51 ± 0.05</td>
<td>3.22 ± 0.01</td>
<td>1.75 ± 0.20</td>
<td>1.66 ± 0.07</td>
<td>1.68 ± 0.10</td>
</tr>
<tr>
<td>Join Date</td>
<td>4.03 ± 0.06</td>
<td>3.73 ± 0.01</td>
<td>2.41 ± 0.45</td>
<td>3.10 ± 0.07</td>
<td>1.70 ± 0.13</td>
</tr>
</tbody>
</table>

Each user has 12 attributes, among which district, age, or join date is used as a confounder \( z \) to build 3 different datasets, with remaining 11 attributes used as covariate \( x \). Treatment \( t \) and outcome \( y \) are synthesised as following:

\[
t \sim \text{Bern}(g(z)); \quad y = t + 10(g(z) - 0.5) + \epsilon, \epsilon \sim \mathcal{N}(0, 1)
\]

Note that district is of 3 categories; age and join date are also discretized into three bins. \( g(z) \) maps these three categories and values to \( \{0.15, 0.5, 0.85\} \).

Some assumptions to justify our method may not hold in this dataset. The important challenges are 1) \( x \) obviously does not satisfy ignorability, and 2) large outcome noise exists. On the other hand, given the huge network structure, most users can practically be identified by their attributes and neighborhood structure, which means \( z \) can be roughly seen as a deterministic function of \( G, x \). Then, \( G, x \) can be, as defined by us, 

noiseless proxies of \( z \) (see Appendix 8.5). CFVAE is then expected to control for the confounding to a large extent and able to learn a balancing score based on
Theorem 3 if we can exploit the network structure effectively. This idea is comparable to Assumptions 2 and 4 in Veitch et al. (2019), which postulate directly that a balancing score can be learned in the limit of infinite large network.

To extract information from the network structure, we use Graph Convolutional Network (GCN) (Kipf & Welling, 2017) in conditional prior and encoder of CFVAE. A difficulty is that, the network G and covariates X of all users are always needed by GCN, regardless of whether it is in training, validation, or testing phase. However, the separation can still make sense if we take care that the treatment and outcome are used only in the respective phase, e.g., \((y_m, t_m)\) of a testing user \(m\) is only used in testing. See Appendix for details.

7 DISCUSSION

In this work, we proposed a new VAE architecture for estimating causal effects under unobserved confounding, with theoretical analysis and state-of-the-art performance. To the best of our knowledge, this is the first generative learning method that provably identifies treatment effects, without directly assuming that the true latent variable can be recovered. It is achieved by, on the one hand, noticing we only need the part of latent information that is correlated to treatment assignment, and, on the other hand, exploiting the recent advances that the latent variable can be recovered up to trivial transformations in a broad class of generative models.

Despite the formal requirement, the experiments show our method is robust to large outcome noise. Theoretical analysis of this phenomenon is an interesting direction for future work. A related theoretical issue is that, while Khemakhem et al. (2020) assumes fixed distribution of noise on \(y\), we observed that, in most cases, allowing the noise distribution to depend on \(z, t\) improves performance. Extending identifiability to conditional noise models is also an interesting direction.

When the latent model is misspecified (Sec. 6.2 and 6.3), our method still matches the state-of-the-art, though we cannot see apparent relationship between recovered latent variable and the true one. It would be nice to see the learned representation indeed preserves causal properties under model misspecification, for example, by some causally-specialized metrics, e.g. Suter et al. (2019). Given the fact that all nonlinear ICA based identifiability requires an injective mapping between the latent and observed variables, theoretical extensions to discrete latent variable would be challenging.

REFERENCES


8 APPENDIX

8.1 Proofs

Proof of Corollary 1. We need the consistency\(^4\) of our VAE to learn a observational distribution equaling to the true one in the limit of infinite data, so that the learned parameters \(\theta_t\) is in the equivalence class of \(\theta\), defined by \((\ref{eq:equivalence_class})\). This can be proved (Khemakhem et al., 2020) Theorem 4) by assuming: 1) our VAE is flexible enough to ensure the ELBO is tight (equals to the log likelihood) for some parameters; 2) the optimization algorithm can achieve the global maximum of ELBO (again equals to the log likelihood).

In this proof, all equations and variables should condition on \(t\), and we omit the conditioning in notation for convenience. In the limit of \(\sigma_y \to 0\), the decoder degenerates to a delta function: \(p(y|z) = \delta(y - f(z))\), we have \(y = f(z)\) and \(y' = f'(z')\). From the consistency of VAE, \(y\) should have the same support as \(y'\). For all \(y\) in the support, there exist a unique \(z\) and a unique \(z'\) satisfy \(y = f(z) = f'(z')\) (use injectivity). Substitute \(y = f(z)\) into the l.h.s of \((\ref{eq:equivalence_class})\), and \(y = f'(z')\) into the r.h.s, we have \(z = \text{diag}(a)z' + b\). The relation is one-to-one for all \(z\), so we get \(z = A(z')\). Similar result for \(f\) follows. □

Proposition 2 (Properties of conditional independence). For random variables \(w, x, y, z\). We have (Pearl, 2009) 1.1.55):

\[
\begin{align*}
\text{x \indep y | z} & \land \text{x \indep w | y, z} \implies \text{x \indep w, y | z} \text{ (Contraction).} \\
\text{x \indep w, y | z} & \implies \text{x \indep y | w, z} \text{ (Weak union).} \\
\text{x \indep w, y | z} & \implies \text{x \indep y | z} \text{ (Decomposition).}
\end{align*}
\]

Proof of Proposition 1. This proof will use the above three properties of conditional independence. We first write our assumptions in conditional independence, as \(A1\). \(t \indep z | x\) (balancing covariate), \(A2\). \(w \indep t | z\) (ignorability given \(z\)), and \(A3\). \(w \indep x | z, t\), where \(w := (y(0), y(1))\).

Now, from \(A2\) and \(A3\), using contraction, we have \(w \indep x, t | z\), then using weak union, we have \(w \indep t | x, z\). From this last independence and \(A1\), using contraction, we have \(t \indep z, w | x\). Then \(t \indep w | x\) follows by decomposition.

Proof of Theorem 2. From the proof of Theorem 1 in Khemakhem et al. (2020), we know \(A_s, b_t\) depend on \(t\) only through \(h, k\). But we assume \(h, k\) do not depend on \(t\). So we have \(z = A(z')\) for both \(t = 0, 1\). From Theorem 1, \(z'\) is a balancing score of \(z\), and satisfies strong ignorability. Here, we proceed a bit different from \((\ref{eq:mu_x})\), we have

\[
\mu_I(x) = \mathbb{E}(\mathbb{E}(y|t)z', x = x) = \int \mathbb{E}(y|t)z' = z', x = x)p(z'|x = x)dz' = \int \mathbb{E}(y|z' = z', x = x, t = t)p(z'|x = x)dz' = \int (\int p(y|z', x, t)d'y)p(z'|x)dz' = (12)
\]

Compare the rightmost side to \((\ref{eq:mu_x})\), note that, there is no conditioning on \(t\) in \(p(z'|x)\), because we use the strong ignorability given \(z'\) (and consistency of counterfactuals) in the third equality, after expanding the outer expectation.

From the consistency of VAE, \(p(y|z', x, t) = \delta(y - f(tz'))\), and \(q(z'|x, y, t) = \delta(z' - r(tz'))\) where \((x, y_t) := (x, y, t)\) is a data point. And \(p(z'|x) = \sum_t \int_y p(z'|x, y, t)p(y, t|x)dy = \sum_t \int_y \delta(z' - f^{-1}_t(y))p(y, t|x)dy\). We have

\[
\mu_I(x) = \int_{z'} (f_t(z')) \sum_t \int_y \delta(z' - f_t^{-1}(yt))p(y, t|x)dydz' = \sum_t \int_y f_t(f_t^{-1}(yt)p(y, t|x)dy = \mathbb{E}(f_t(r_t(x, y_t))|x = x)
\]

\(^4\)This is the statistical consistency of an estimator. Do not confuse with the consistency of counterfactuals.
We should note that \( p(z'|x, y, t) := p_{\theta'}(z'|x = x, y = y, t = t) = p_{\theta'}(y = y, z'|x = x, t = t) / \int p_{\theta'}(y = y, z'|x = x, t = t) dz' \) might not be equal to the truth \( p(z|x, y, t) := p_{\theta}(z|x = x, y = y, t = t) \). Similar situations prevail when identifiability is achieved by nonlinear ICA (Hyvärinen & Morioka, 2016; Hyvärinen et al., 2019; Khemakhem et al., 2020).

Also note that the learning of inverse mapping \( f_i^{−1} \) in the encoder \( q \) is enforced by consistency \((q(z'|x, y, t) = p(z'|x, y, t))\), we can just use an MLP for \( r_i' \) in the encoder, and we will have \( r_i' = f_i^{−1} \) if the MLP is flexible enough to contain \( f_i^{−1} \). Similar situations prevail when identifiability is achieved by nonlinear ICA (Hyvärinen & Morioka, 2016; Hyvärinen et al., 2019; Khemakhem et al., 2020).

8.2 On the three identification conditions

Ignorability given \( z \) means there is no correlation between factual assignment of treatment and counterfactual outcomes given \( z \), just as it is the case in RCT. Thus, it can be understood as unconfoundedness given \( z \), and \( z \) can be seen as the confounder(s) we want to control for. Positivity says the supports of \( p(t = 1|x = x) \), \( t = 0,1 \) should be overlapped, and this ensures there are no impossible events in the conditions after adding \( t = t \), and the expectations can thus be estimated from observational data. Finally, consistent counterfactuals are well defined: given assignment of treatment \( t = t \), the observational outcome \( y \) should take the same value as the potential outcome \( y(t) \).

8.3 Conditional VAE

By adding a conditioning variable \( c \) (usually a class label), Conditional VAE (CVAE) (Sohn et al., 2015; Kingma et al., 2014) can give better reconstruction of observation of each class. The variational lower bound is

\[
\log p(y|c) \geq E_{z \sim q} \log p(y|z, c) - D_{KL}(q(z|y, c)||p(z|c)) := L_{CV_AE}(y, c) \tag{14}
\]

The conditioning on \( c \) in the prior is usually omitted, since the dependence between \( c \) and the latent representation is also involved in the encoder \( q \).

8.4 Identifiability of representation in sec. 5.1 is not enough

Consider how the recovered \( z' \) would be used. For a control group \( (t = 0) \) data point \( (x, y, 0) \), the real challenge is to predict the counterfactual outcome \( y(1) \). Taking the observation, the encoder will output a posterior sample point \( z_0' = f_0^{−1}(y) = A_0^{-1}(z_0) \) (with zero outcome noise, the encoder degenerates to a delta function: \( q(z|x, y, 0) = \delta(z - f_0^{−1}(y)) \)). Then, we should do counterfactual inference, using decoder with counterfactual assignment \( t = 1 \): \( y_1' = f_1'(z_0') = f_1 \circ A_1(A_0^{-1}(z_0)) \). This prediction can be arbitrary far from the truth \( y(1) = f_1(z_0) \), due to the difference between \( A_1 \) and \( A_0 \). More concretely, this is because when learning the decoder, only the posterior sample of the treatment group \( (t = 1) \) is fed to \( f_1' \), and the posterior sample is different to the true value by the affine transformation \( A_1 \), while it is \( A_0 \) for \( z_0' \).

8.5 Two special cases of balancing covariate

Definition 2 (Noiseless proxy). Random variable \( x \) is a noiseless proxy of random variable \( z \) if \( z \) is a function of \( x \) (\( z = \omega(x) \)).

Noiseless proxy is a special case of balancing covariate because if \( x = x \) is given, we know \( z = \omega(x) \) and \( \omega \) is a deterministic function, then \( p(z|x = x) = p(z|x = x, t) = \delta(z - \omega(x)) \). Also note that, a noiseless proxy always has higher dimensionality than \( z \), or at least the same.

Intuitively, if the value of \( x \) is given, there is no further uncertainty about \( z \), so the observation of \( x \) may work equally well to adjust for confounding. But, as we will see soon, a noiseless proxy of the true confounder does not satisfy positivity.

Definition 3 (Injective proxy). Random variable \( x \) is an injective proxy of random variable \( z \) if \( x \) is an injective function of \( z \) (\( x = \chi(z), \chi \) is injective).
Injective proxy is again a special case of noiseless proxy, since, by injectivity, \( z = \chi^{-1}(x) \), i.e. \( z \) is also a function of \( x \).

Under this very special case, that is, if \( x \) is an injective proxy of the true confounder \( z \), we finally have \( x \) is a balancing score and satisfies strong ignorability, since \( x \) is a balancing covariate and a function of \( z \). To see this in another way, let \( f = e \circ \chi^{-1} \) and \( b = \chi \) in Theorem 1, then \( f(x) = f(b(z)) = e(z) \). By strong ignorability of \( x \), \( \bar{E}(y(x)) = \bar{E}(y|x = x, t = t) \). Thus, a regression of \( y \) on \( (x, t) \) will give a valid estimator of CATE and ATE.

However, a noiseless but non-injective proxy is not a balancing score, in particular, positivity might not hold. Here, a simple regression will not do. This is exactly because \( \omega \) is non-injective, hence multiple values of \( x \) that cause non-overlapped supports of \( p(t = t|x = x), t = 0, 1 \) might be mapped to the same value of \( z \). An extreme example would be \( t = I(x > 0), z = |x| \). We can see \( p(t = t|x) \) are totally non-overlapped, but \( \forall t, z \neq 0 : p(t = t|z = z) = 1/2 \).

8.6 Details and additional results for experiments

8.6.1 Synthetic data

| \( p(x|w) \) | \( p(y|x, t) \) | \( p(x|z) \) | \( p(y|x, t) \) | \( p(x|z) \) | \( p(y|x, t) \) |
|---|---|---|---|---|---|
| \( w \) | \( z \) | \( y \) | \( x \) | \( t \) | \( y \) |

Figure 4: Graphical models for generating synthetic datasets. From left: IV, ignorability given \( x \), and non-balancing proxy \( x \). Note that in the latter two cases, reversing the arrow between \( x, z \) does not change any independence relationships, and causal interpretations of the graphs remain the same.

Interestingly, linear outcome models seem harder for both methods, maybe because the two true linear outcome models for \( t = 0, 1 \) are more similar, and it is harder to distinguish and learning outcome models. Note that after generating the outcomes and before the data is used, we normalize the distribution of ATE of the 100 generating models, so the errors on linear and nonlinear settings are basically comparable.

You can find more plots for latent recovery at the end of the paper.

8.6.2 IHDP

IHDP is based on an RCT where each data point represents a child with 25 features about their birth and mothers. Race is introduced as a confounder by artificially removing all treated children with nonwhite mothers. There are 747 subjects left in the dataset. The outcome is synthesized by taking the covariates (features excluding race) as input, hence ignorability holds given the covariates. Following previous work, we split the dataset by 63:27:10 for training, validation, and testing.

8.6.3 Pokec

GCN takes the network matrix \( G \) and the whole covariates matrix \( X := (x^T_1, \ldots, x^T_M)^T \), where \( M \) is user number, and outputs a representation matrix \( R \), again for all users. During training, we select the rows in \( R \) that correspond to users in training set. Then, treat this training representation matrix as if it is the covariates matrix for a non-networked dataset, that is, the downstream networks in conditional prior and encoder are the same as in the above two experiments, but take \( (R_{m:t})^T \) where \( x_m \) was expected as input. And we have respective selection operations for validation and testing.
We can still train CFVAE including GCN by Adam, simply setting the gradients of non-selected rows of $R$ to 0.

Note that GCN cannot be trained using mini-batch, instead, we perform batch gradient decent using full dataset for each iteration, with initial learning rate $10^{-2}$. We use dropout (Srivastava et al., 2014) with rate 0.1 to prevent overfitting.

The pre-treatment $\sqrt{\text{PFEHE}}$ for Age, District, and Join date confounders are 1.085, 0.686, and 0.699 respectively, practically the same as the ATE errors.

8.7 Additional plots on synthetic datasets

See next pages.
Figure 6: Plots of recovered-true latent under unobserved confounding. Rows: first 10 nonlinear random models, columns: proxy noise level.
Figure 7: Plots of recovered-true latent under unobserved confounding. Rows: first 10 nonlinear random models, columns: outcome noise level.
Figure 8: Plots of recovered-true latent when ignorability holds. Rows: first 10 nonlinear random models, columns: proxy noise level.
Figure 9: Plots of recovered-true latent when ignorability holds. Rows: first 10 nonlinear random models, columns: outcome noise level.
Figure 10: Plots of recovered-true latent when ignorability holds. Conditional prior depends on $t$. Rows: first 10 nonlinear random models, columns: outcome noise level. Compare to the previous figure, we can see the transformations for $t = 0, 1$ are not the same, confirming our Theorem 3.
Figure 11: Plots of recovered-true latent on IVs. Rows: first 10 nonlinear random models, columns: outcome noise level.