

Hi-CI: Deep Causal Inference in High Dimensions

Ankit Sharma

TCS Research, India

ANKIT.SHARMA16@TCS.COM

Garima Gupta

TCS Research, India

GUPTA.GARIMA1@TCS.COM

Ranjitha Prasad

Indraprastha Institute of Information Technology, Delhi, India

RANJITHA@IIITD.AC.IN

Arnab Chatterjee

TCS Research, India

ARNAB.CHATTERJEE4@TCS.COM

Lovekesh Vig

TCS Research, India

LOVEKESH.VIG@TCS.COM

Gautam Shroff

TCS Research, India

GAUTAM.SHROFF@TCS.COM

Editor: Thuc Duy Le, Lin Liu, Kun Zhang, Emre Kiciman, Peng Cui, and Aapo Hyvärinen

Abstract

We address the problem of counterfactual regression using causal inference (CI) in observational studies consisting of high dimensional covariates and high cardinality treatments. Confounding bias, which leads to inaccurate treatment effect estimation, is attributed to covariates that affect both treatments and outcome. The presence of high-dimensional covariates exacerbates the impact of bias as it is harder to isolate and measure the impact of these *confounders*. In the presence of high-cardinality treatment variables, CI is rendered ill-posed due to the increase in the number of counterfactual outcomes to be predicted. We propose **Hi-CI**, a deep neural network (DNN) based framework for estimating causal effects in the presence of large number of covariates, and high-cardinal and continuous treatment variables. The proposed architecture comprises of a decorrelation network and an outcome prediction network. In the decorrelation network, we learn a data representation in lower dimensions as compared to the original covariates, and addresses confounding bias alongside. Subsequently, in the outcome prediction network, we learn an *embedding* of high-cardinality and continuous treatments, jointly with the data representation. We demonstrate the efficacy of causal effect prediction of the proposed **Hi-CI** network using synthetic and real-world NEWS datasets.

Keywords: Causal Inference, Confounding, High-dimensions, Deep neural networks, Embedding

1. Introduction

In our daily lives, humans often justify several actions and events in terms of *cause and effect* and incorporate the feedback into future actions in order to alter future outcomes. On the other hand, supervised learning based machine learning techniques focus on predicting outcomes, where outcomes are strongly tied to the nature of training data. However, it is

possible that when this model is used in real-life scenarios, data generating process may vary vastly, and hence these models do not generalize well. Hence, there have been several efforts by researchers to integrate causality into machine learning models for obtaining robust and generalizable machine learning models. It is well-accepted that obtaining causal relations from an observational dataset is possible if underlying data generating process is well-understood. This is often posed as a problem of predicting the effects of interventions (or *treatments*) in the data generating process, and such treatments are generally enforced using policy or operational changes. Furthermore, understanding the effect of intervention requires us accurately answer counterfactual or *what-if* type questions, which in turn necessitates modelling the causal relationship between the treatment and outcome variables.

Causal inference (CI) for observational studies lies at the heart of various domains like healthcare, digital marketing, econometrics based applications, etc, that require quantifying the effect of a treatment or an intervention on an individual. As an example, consider a retail outlet optimizing the waiting time at a store since long queues leads to loss in customer base, in turn leading to low sales. In their historical observational data, consider the queue-length as a treatment variable and sale as an outcome variable. First, note that queue-length varies in the training data since it depends on the number of items purchased by every customer. A discount sale leads to a given customer buying more leading to higher queue-length. That is, in our training set we observe examples with long queues and high sales. A naive supervised learning approach might incorrectly predict that increase in queue-length leads to increase in sales, whereas the true relationship between queue-length and sales is surely negative on regular days. Typically we have information regarding discount sales, and including them in the model can correct for such effects. Such variables affect both, the outcome and the treatment, and hence, these variables are known as *confounding covariates* in the CI problem. Similarly, in a digital marketing context, *age* can be a confounding covariate which introduces selection bias in providing advertisements to young, middle-aged and old-aged users and consequently a varying buying behaviour (outcome). These aspects as well-captured in Simpson’s paradox (Bottou et al., 2013), which states that the confounding behavior may lead to erroneous conclusions about causal relations and counterfactual estimation when the confounding variable is not considered in analysis.

A key problem in modern empirical work is that datasets consists of large numbers of covariates (Newman, 2012) and high-cardinality treatments (Diemert et al., 2017). We motivate this using the following real-world scenarios

1. High-dimensional covariates: A typical characteristic of genomic data is the presence of vast number of covariates. For example, a problem of interest is to genetically modify the plant *Arabidopsis thaliana* to shorten the time to flowering (Bühlmann, 2013) since fast growing crops lead to better food production. In the corresponding dataset, there are 47 instances of the outcome *time to flowering* and 21,326 genes which are construed as *covariates*. The goal is to causally infer the effects of a single gene intervention on the outcome, considering the other genes as the covariates. A similar (but less severe) situation is also seen in the popular *The Cancer Genomic Atlas* (TCGA) project (Weinstein et al., 2013) which is a repository that consists of gene expression values of 20547 genes of 9659 individuals. Here the goal is to measure the gene expression values for several treatment strategies like medication,

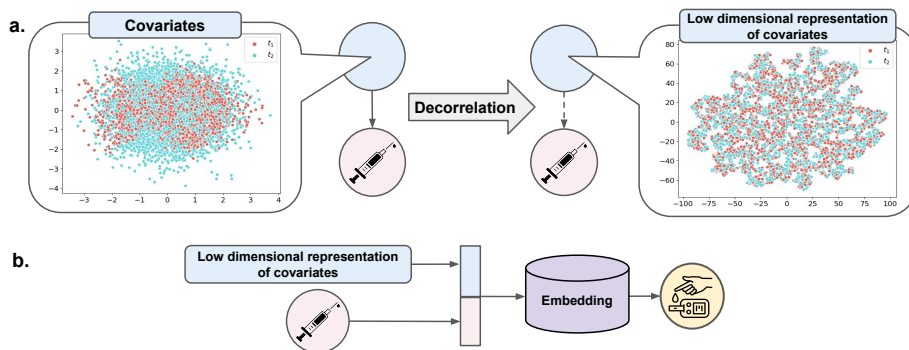


Figure 1: **a.** The t-SNE plot on the left and right depicts the decorrelated transformation of high-dimensional covariates in data, into a low-dimensional representation, using the Hi-CI framework. **b.** Illustrates the dosage embedding to learn a low-dimensional representation of treatments followed by outcome prediction in the Hi-CI framework.

chemotherapy and surgery (Schwab et al., 2019), so that the best treatment regimen is chosen.

2. High-cardinality treatments: We provide the example of the Criteo dataset to motivate high cardinality treatments. Criteo dataset (Diemert et al., 2017) includes browsing related activities of users for interaction with 675 campaigns. In the causal setting, these campaigns are considered as *treatments* with *campaign effect on buying* as the outcome (Dalessandro et al., 2012).
3. High dimensional covariates, high cardinality treatments with dosages: The popular NEWS datasets consists of news items represented by 2870 bag-of-word covariates. These news items are read by viewers on media devices. In causal setting, media devices act as treatments. Since the number of news items can vary from few tens to hundreds, varying but finite viewing time is considered as dosage levels, while the readers' opinion on different media devices is considered as outcome (Schwab et al., 2019).

In the above applications of healthcare, advertising etc, an individual's response plays an important role in guiding practitioners to select the best possible interventions. Hence, it is essential to build models to handle such high dimensional scenarios. This motivates us to design machine learning models that abate confounding effects, while being parsimonious in representation of high-dimensional variables, and adequately flexible.

Related Work: Treatment effect estimation in the presence of high dimensional covariates is a well-explored topic in statistical literature on causal inference. In (Robins et al., 1994), the authors propose techniques based on inverse probability of treatment weighting (IPTW), which is sensitive to the propensity score model (Fan et al., 2016). Propensity score estimation is improved by employing covariate balancing propensity scores (CBPS) in high dimensions (Imai and Ratkovic, 2014; Guo et al., 2016; Fan et al., 2016). LASSO regression

for high dimensional CI was proposed in (Belloni et al., 2014). Approximate residual balancing techniques for treatment effect estimation in high dimensions is proposed in (Athey et al., 2018). A common trait among these works is that they focus on estimating the average treatment effect (ATE) in the presence of a large number of covariates, but are limited to settings with only two treatments. In (Schwab et al., 2019), high-cardinality treatments and continuous treatments are considered. Typically, in the context of continuous treatments, a given treatment is represented using *multiple dosage levels* (Schwab et al., 2019) to account for the exploding cardinality of the treatment set (as each dosage is a unique treatment in itself). In statistical literature, continuous dosages have been handled using propensity scores (Hirano and Imbens, 2004), doubly robust estimation methods (Kennedy et al., 2017), generalized CBPS score (Fong et al., 2018), using estimation frameworks for both treatment assignment and outcome prediction (Galagate, 2016).

Modern deep neural networks (DNN) based methods employ matching or balancing techniques for compensating confounding bias. Existing DNN based architectures for the multiple treatment scenario as proposed in (Sharma et al., 2020; Schwab et al., 2018) have a severe limitation with respect to their architectures. They employ a separate regression network per treatment, and hence, these neural networks cannot be used in the presence of a large number of treatments. Furthermore, in the presence of high-dimensional covariates, it is essential to design a parsimonious, yet lossless representation of these covariates. In several works such as (Johansson et al., 2016; Shalit et al., 2017), a latent representation for covariates is learned by minimizing the discrepancy distances of the control and treatment populations to compensate for confounding bias, in the presence of binary treatments. Since such a data representation is not lossless, this approach is not suitable in the presence of high-cardinality variables. An autoencoder is used to learn an unbiased lossless representation of covariates, uncorrelated with respect to the multiple, yet small number of treatment variables (Atan et al., 2018; Zhang et al., 2019). On the other hand, matching based DNN techniques find similar individuals with dissimilar treatments using propensity scores (Schwab et al., 2018; Sharma et al., 2020; Ho et al., 2007). Matching is often accomplished using nearest neighbour match (Ho et al., 2007), propensity score (Schwab et al., 2018) or generalized propensity score (Sharma et al., 2020). These techniques are computationally infeasible in the presence of high-cardinality treatment variables as good recipes for matching require spanning the entire dataset in search of alternate treatment variables while ensuring a balance in the number of individuals per treatment. However, the success of the DNN based causal CI techniques and vacuum in the literature on architectures that handle large number of covariates and treatments in observational datasets motivates us to address this problem.

Contributions: In the presence of high-cardinality treatment variables, the number of counterfactual outcomes to be estimated is much larger than the number of factual observations, rendering the problem to be ill-posed. Furthermore, lack of information regarding the confounders among large number of covariates pose challenges in handling confounding bias. Hence, it becomes essential to find a lower dimensional manifold where an equivalent problem of causal inference can be posed, and counterfactual outcomes can be computed. To this end, we propose a DNN based **Hi-CI**, specifically designed to measure causal effects in the presence of countably large number of covariates and high-cardinality treatments. We propose a novel loss function which is designed to achieve the following goals:

- Obtain an autoencoder based data representation for high-dimensional covariates while simultaneously handling confounding bias using a decorrelation loss, as depicted in (a) of Fig. 1.
- Caters to both, a large number of discrete, and continuous treatments, where a continuous treatment is characterised by a fixed number of dosage levels.
- Obtain a per-dosage level embedding layer to learn the low-dimensional representation of the high-cardinality treatments by jointly training the network using root mean square (RMSE) loss and a sparsifying mixed norm loss function as depicted in lower part (b) of Fig. 1.

We demonstrate that the proposed Hi-CI architecture outperforms the state-of-the-art CI techniques such as perfect match (PM) (Schwab et al., 2018), MultiMBNN (Sharma et al., 2020), Deep-Treat (Atan et al., 2018) and DRNet (Schwab et al., 2019) in the presence of high-dimensional variables. We use precision in estimation of heterogeneous effect (*PEHE*) and mean absolute percentage error (*MAPE*) over the ATE as metrics in all the experiments. For treatments with dosages, we use mean integrated square error (*MISE*) as a metric, and in addition, we also introduce a new metric for computing *MAPE* over the dosage effect.

2. CI in High dimensions: Preliminaries and Loss Function

In this section, we describe the causal inference problem, especially in the presence of a large number of covariates and treatments. We motivate the need for finding a low-dimensional representation of the high-cardinality covariates and treatments and describe the loss function that helps to learn these representations from data and hence provide the basis for understanding the Hi-CI DNN described in the sequel.

2.1 Causal Inference Preliminaries

We consider training data \mathbf{D}_{CI} , comprising of N samples from an observational dataset, where each sample is given by $\{x_n, \mathbf{t}_n, \mathbf{y}_n\}$, where $x_n \in \mathbf{X}$. Each individual (also called *context*) n is represented using P covariates, i.e., $x_{n,p}$ denotes the p -th covariate of the n -th individual, for $1 \leq n \leq N$. Furthermore, an individual is subject to one of the K treatments given by $\mathbf{t}_n = [t_n(1), \dots, t_n(K)]$, where each entry of \mathbf{t}_n is binary, i.e., $t_n(k) \in \{0, 1\}$. Here, $t_n(k) = 1$ implies that the k -th treatment is provided. We assume that only one treatment is provided to an individual at any given point in time, and hence, \mathbf{t}_n is an one-hot vector. A *counterfactual* is defined based on $K - 1$ alternate treatments, and corresponding outcomes are referred to as counterfactual outcomes. Accordingly, the response vector for the n -th individual is given by $\mathbf{y}_n \in \mathbb{R}^{K \times 1}$, i.e., the outcome is a continuous random vector with K entries denoted by $y_n(k)$, the response of the n -th individual to the k -th treatment. The set of counterfactual responses for the n -th individual comprises of response to treatments $l \neq k$, given by $y_{n,l}$, and the size of this set is $K - 1$. In the case of continuous treatment, we assume that $t_n(k) \in \mathbb{R}$ which implies that $\mathbf{t}_n \in \mathbb{R}^{K \times 1}$, that is the treatment is a real-valued vector. However, for ease of representation, we cast the continuous treatment variable using a finite set of E dosage levels where E remains constant across treatments. Following the

notation for discrete treatments, the outcome is a continuous random vector denoted by $y_n(k_e)$, where $1 \leq k_e \leq KE$, is the response of the n -th individual to the e -th dosage level of the k -th treatment. In the case of discrete treatment, the maximum size of outcomes to be predicted by the DNN is $N(K - 1)$, while the number of available factual outcomes are N in number. It is evident that this problem is ill-posed when K is large. Furthermore, in the case of continuous treatments, we effectively have KE treatments, leading to $N(KE - 1)$ counterfactual responses.

In this work, we consider observational studies where there are large number of covariates P and large number of treatments K . Our goal is to train a DNN model to overcome confounding and perform counterfactual regression, i.e., to predict the response given any context and treatment, for large P and K . In the sequel, we describe different components of the the loss function that tackles confounding bias, high-dimensional treatments and high-dimensional covariates.

2.2 Learning Representations from Data

The crux of the loss function in CI for observational studies lie in techniques employed to compensate for the confounding bias. In this direction, we employ autoencoders which simultaneously encourage confounding bias compensation and learning compressed representation for the high-dimensional covariates. Alongside, we employ RMSE with mixed-norm regularizer based loss-function to obtain a low-dimensional representation for treatments. In the sequel, we describe the mathematical constructs of learning the representation and the loss function.

2.2.1 DATA REPRESENTATION USING AUTOENCODERS

In this work, we propose to use the autoencoder to jointly obtain a low-dimensional representation of the high-dimensional covariates and alleviate the effect of confounding. Let \mathbf{T} represent the set of treatments, and $T_k \in \mathbf{T}$ be a random variable; instantiation for the n -th individual is $t_n(k)$. Using an autoencoder, we seek a mapping from the space of covariates $\mathbf{X} \in \mathbb{R}^{N \times P}$, such that $\Phi : \mathbf{X} \rightarrow \mathcal{R}$, where $\mathcal{R} \in \mathbb{R}^{N \times L}$ is the representation space. The mapping Φ is such that,

- The induced distribution of the treatments over \mathcal{R} , which we denote $p(T_k | \Phi(\mathbf{X}))$ is free of confounding bias for all k .
- The representation of \mathbf{x}_n under $\Phi(\cdot)$ for all n is lossless.
- It maps higher dimensional covariates in P to a low-dimensional space of size L , i.e., $L < P$.

A typical propensity score based matching approach addresses the issue of confounding bias by balancing the propensity score to obtain similar covariate distributions across treated populations. Mathematically, a sub-sample \mathbf{X}_s of the original sample is considered such that it ensures that the following condition holds:

$$p(T_1 | \mathbf{X}_s) = p(T_2 | \mathbf{X}_s) = \dots = p(T_K | \mathbf{X}_s). \quad (1)$$

Note that the condition stated above does not necessitate that treatment and covariates variables are uncorrelated. On the other hand, the loss function associated to the autoencoder imposes a far more stringent condition (Atan et al., 2018) such that

$$p(T_k|\mathbf{X}) = p(T_k), \quad \forall k, \quad (2)$$

for the entire sample \mathbf{D}_{CI} . Autoencoders have been employed in the literature for addressing some of the tasks such as lossless data representation (Atan et al., 2018; Ramachandra, 2018). However, to the best of the authors' knowledge, this is the first-of-its-kind approach where an autoencoder is used to jointly accomplish the goals as specified above, and primarily, low-dimensional representations.

To ensure lossless data representation, the loss function associated with the autoencoder jointly minimizes the mean-squared error loss between the reconstructed and the original covariates, and the distance between the unbiased ($p(T_k)$) and the biased treatment distributions ($p(T_k|\Phi(\mathbf{X}))$) for all k , while maintaining the resultant mapping in a lower dimension as compared to the original covariates ($L < P$). These goals can be achieved by using the following loss function:

$$\mathcal{L}_1(\Phi, \Psi, \beta) = \mathcal{L}_{ce}(\Phi(\mathbf{X})) + \beta \mathcal{L}_{ae}(\Phi(\mathbf{X}), \Psi(\Phi(\mathbf{X}))), \quad (3)$$

where $\mathcal{L}_{ce}(\Phi)$ is the cross-entropy measure. The cross-entropy loss is directly proportional to the Kullback-Liebler divergence between the distributions in question, and hence it is an appropriate metric to minimize the divergence between $p(T_k)$ and $p(T_k|\Phi(\mathbf{X}))$ for all k . Accordingly, $\mathcal{L}_{ce}(\Phi)$ is given by

$$\mathcal{L}_{ce}(\Phi) = - \sum_{T \in \mathbf{T}} p(T) \log(p(T|\Phi(\mathbf{X}))). \quad (4)$$

Furthermore, the loss term $\mathcal{L}_{ae}(\Phi, \Psi)$ is employed to minimize the mean-squared loss between the reconstructed and the original covariates in the autoencoder. Mathematically,

$$\mathcal{L}_{ae}(\Phi, \Psi) = \frac{1}{PN} \sum_{n=1}^N \sum_{p=1}^P \|(\mathbf{x}_{n,p} - (\Phi \circ \Psi)(\mathbf{x}_{n,p}))\|^2, \quad (5)$$

where Ψ is the decoder mapping such that $\Psi : \mathcal{R} \rightarrow \mathbf{X}$ and \circ is a composition operator, and $L < P$, which ensures that we obtain a low-dimensional, yet meaningful representation of the high-dimensional covariates.

As a regularizer, we propose to employ the mixed norm on the difference of means, represented using the matrix \mathbf{M}_D . The columns of \mathbf{M}_D are given by $\mu_{D,(T_i,T_j)} = \frac{1}{LK(K-1)}(\mu_{T_i}(\Phi(\mathbf{X})) - \mu_{T_j}(\Phi(\mathbf{X})))$, where $\mu_{T_i}(\Phi(\mathbf{X})) \in \mathbb{R}^{L \times 1}$ is the mean of representation for all individuals in \mathbf{X} , given by $\Phi(\mathbf{X})$, that undergo treatment T_i . Since all possible pairs of treatments (T_i, T_j), for all T_i and T_j are considered, \mathbf{M}_D is of dimension $\mathbb{R}^{L \times (K(K-1))}$. The mixed norm regularizer on \mathbf{M}_D , denoted as $\mathcal{L}_{2,1}(\mathbf{M}_D)$, is as follows:

$$\mathcal{L}_{2,1}(\mathbf{M}_D) = \sum_{u=0}^{K(K-1)} \sqrt{\sum_{v=0}^{L-1} |M_D(u, v)|^2}. \quad (6)$$

Combining the above mentioned loss functions, we obtain the following loss function from the decorrelating network:

$$\mathcal{L}_D(\Phi, \Psi, \beta, \gamma) = \mathcal{L}_{ce}(\Phi) + \beta \mathcal{L}_{ae}(\Phi, \Psi) + \gamma \mathcal{L}_{2,1}(\mathbf{M}_D). \quad (7)$$

The above objective function cannot be computed directly since both $p(T_k|\Phi(\mathbf{X}))$ and $p(T_k)$ are unknown for any k . The estimates of $p(T_k)$ for $1 < k \leq K$ is given by (Atan et al., 2018):

$$p(T_k = t) = \frac{\sum_{n=1}^N \mathbb{I}(t_n(k) = t)}{N}, \quad (8)$$

where $\mathbb{I}(\cdot)$ is the indicator function. Essentially, $p(T_k)$ provides a count-based probability of k -th treatment. Further, the functional form of $p(T_k|\Phi(x_n))$ is assumed to be similar to logistic regression:

$$p(T_k|\Phi(\mathbf{x}_n)) = \frac{\exp((\boldsymbol{\theta}_{T_k})^T \Phi(\mathbf{x}_n))}{\sum_{k=1}^K \exp((\boldsymbol{\theta}_{T_k})^T \Phi(\mathbf{x}_n))}, \quad (9)$$

where $\boldsymbol{\theta}_{T_k} \in \mathbb{R}^{L \times 1}$ are the per-treatment parameters of the logistic regression framework. This results in

$$\mathcal{L}_{ce}(\Phi) = - \sum_{k=1}^K p(T_k) (\boldsymbol{\theta}_{T_k})^T \Phi(\mathbf{x}_n) - \log \left(\sum_{k=1}^K (\boldsymbol{\theta}_{T_k})^T \Phi(\mathbf{x}_n) \right). \quad (10)$$

2.2.2 EMBEDDINGS FOR HIGH-DIMENSIONAL TREATMENT

As mentioned in the previous subsection, the goal of this work is to design DNN based CI model for datasets with large number of unique treatments. While a single bit is sufficient to represent binary treatments (Johansson et al., 2016), a one hot representation is used within the DNN to represent a categorical treatment for a given user (Sharma et al., 2020). In the presence of high-cardinality treatment variables, i.e., treatments with several unique categories, the size of the one-hot vector becomes unmanageable. Furthermore, DNN architectures that cater to multiple treatments often use a sub-divided network as in (Schwab et al., 2018) and (Schwab et al., 2019), with one branch per treatment. Such a branching network based DNN architecture becomes computationally intractable as the number of treatments increase.

An aspect that matters the most about one-hot encoding is the fact that one-hot mapping does not capture any similarity in treatment categories. For instance, if treatments t_1 and t_2 are drugs for lung-related issues, and t_3 is a treatment for skin-acne which is seemingly an unrelated issue, t_1 , t_2 and t_3 are equidistant in the one-hot encoding space. We propose to learn a representation of treatments denoted as $\Omega : [\Phi(\mathbf{X}), \mathbf{T}] \rightarrow \mathbf{Y}$, where \mathbf{Y} represents the space of output response vectors of length K , and the *embedding* Ω encapsulates closeness property of treatments. Such representations of the treatment space is extremely relevant in the current day observational studies, as explained in the introduction.

The impact of the embedding is realised in the outcome prediction part of the network. The loss on the outputs of the outcome prediction layer is the root mean square error

(RMSE) loss given by

$$\mathcal{L}_{RMSE}(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K \|y_n(k) - \hat{y}_n(k)\|^2} \quad (11)$$

where $\hat{\mathbf{y}}_n = \Omega([\Phi(\mathbf{x}_n), \mathbf{t}_n]^T)$.

Although the impact of embedding is evident only in the above loss function, note that the training of the Hi-CI framework incorporates all of the loss functions combined in (7) and (11). Intuitively, through the mixed norm based regularizer in (6), we minimize the distance between multiple populations whose covariate information is summarized by $\Phi(\mathbf{X})$ and hence, we are able to exploit the similarity properties in the treatment itself. When we train the network using (11) along with (6), in addition to promoting parsimonious representations owing to similarity of treatments, we also ensure that such representation leads to a response close, in the sense of RMSE, to the true label.

2.3 Modified Loss Function when $E > 1$

In the case of continuous treatment, we represent a treatment as consisting of multiple dosages (Schwab et al., 2019). In particular, we assume that each treatment is specified by a set of E dosage levels, i.e., E remains constant across treatments. In the design of DNN, we assume that the treatment is affected by the confounding bias, but the dosage administered is not. However, since we need to infer the per-dosage level counterfactual, we exploit the dosage information available in the labels $\mathbf{y}_n(k_e)$ outcome prediction network. Accordingly, we incorporate the dosage levels in a generalised RMSE loss function:

$$\mathcal{L}_{RMSE}(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K \sum_{e=1}^E \|y_n(k_e) - \hat{y}_n(k_e)\|^2}, \quad (12)$$

where $\hat{\mathbf{y}}_n = \Omega_e([\Phi(\mathbf{x}_n), \mathbf{t}_n]^T)$. Note that if we set $E = 1$ in (12), we obtain (11). Hence, we essentially aggregate the characteristics of all observed individuals subjected to any treatment, and having been administered a given dosage e_n .

In order to implement the loss function associated with continuous treatments, it is also essential to make changes in the network architecture. In the sequel, we describe the proposed DNN architecture of Hi-CI in tandem with components of the loss-function as described in (11) and (7). We also discuss the changes in the network architecture due to the generalised RMSE loss in (12).

3. Hi-CI: Proposed Deep Neural Network Architecture

In this section, we describe the novel Hi-CI network architecture proposed for CI in high-dimensions. The network consists of two components, a decorrelation network and a regression (or outcome prediction) network. The decorrelation network consists of the autoencoder and the corresponding loss function is given in (7). The loss function pertaining to the regression network is given by (11). The overall loss function corresponding to the

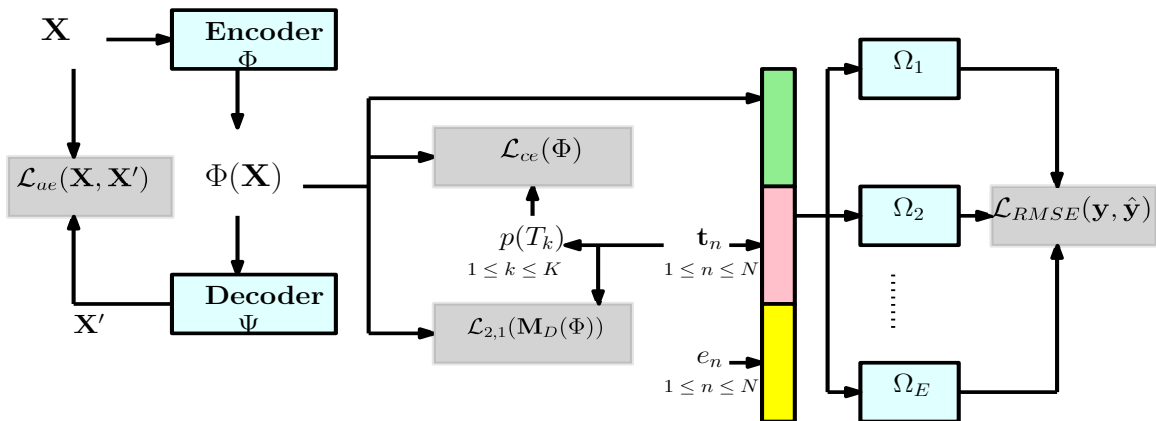


Figure 2: Hi-CI: Proposed DNN architecture for inference in the presence of high-cardinality covariates and continuous treatments. The blocks in cyan represent the parameters to be tuned, and the blocks in grey denote the loss functions.

decorrelation network and regression network is given by

$$\mathcal{L}(\Phi, \Psi, \Omega, \beta, \gamma, \lambda) = \mathcal{L}_D(\Phi, \Psi, \beta, \gamma) + \lambda \mathcal{L}_{RMSE}(\mathbf{y}, \hat{\mathbf{y}}) \quad (13)$$

As explained in the previous section, in the case of continuous treatments, we modify the structure of the regression network alone. In the regression network, we obtain the per-dosage level embedding, which we denote as $\Omega_e(\cdot)$, where $1 \leq e \leq E$. We use the concatenation of learned representation $\Phi(\mathbf{x}_n)$, treatment vector \mathbf{t}_n as an input to the embedding layer. The dosage information is used to obtain a subdivided network, i.e., in this work we split the network based on dosages and not treatments since $E \ll K$. The overall loss function for continuous treatments is given by

$$\mathcal{L}(\Phi, \Psi, \Omega_e, \beta, \gamma, \lambda) = \mathcal{L}_D(\Phi, \Psi, \beta, \gamma) + \lambda \mathcal{L}_{RMSE}(\mathbf{y}, \hat{\mathbf{y}}) \quad (14)$$

The generalized architecture of the Hi-CI network with continuous treatments is as depicted in Fig. 2 (details in Supplementary Information). For discrete treatments, $E = 1$, and hence one embedding sub-network ($\Omega(\cdot)$) is used instead of multiple sub-networks ($\Omega_e(\cdot)$) for outcome prediction.

4. Experimental Set-up

In this section, we describe the experimental setup to demonstrate the efficacy in counterfactual regression of the Hi-CI algorithm and report the results obtained. We report the results on a synthetically generated dataset (Sun et al., 2015), and the semi-synthetic NEWS dataset (Johansson et al., 2016) for evaluation.¹ Since a counterfactual outcome is not available, it becomes impossible to test CI algorithms in the context of counterfactual prediction. As a solution, data generating processes (DGP) are employed for demonstrating

1. The simulated datasets will be available upon request from authors post publication of the paper.

the results. In this section, we describe the datasets employed as well as the corresponding DGPs employed for each dataset. Furthermore, we describe the metrics we use for evaluating the Hi-CI framework where $E = 1$, namely precision in estimation of heterogeneous effect (PEHE) (Shalit et al., 2017) and Mean Absolute Percentage Error (MAPE) over Average Treatment Effect (ATE) (Sharma et al., 2019). In the case of continuous treatments, i.e., for $E > 1$, we evaluate the Hi-CI framework using Mean Integrated Squared Error (MISE) and MAPE over ATE with dosage metric.

4.1 Datasets

In this subsection, we describe the datasets and the DGP employed for each dataset.

- **Synthetic (Syn):** We use a synthetic process described in (Sun et al., 2015) to generate data for both multiple treatment as well as continuous valued treatment scenario. The DGP gives us the flexibility to simulate the counterfactual responses along with the factual treatments and responses, thereby helping in better evaluation of the proposed model. The generation process in (Sun et al., 2015) allows for 5 confounding covariates while the remaining $P-5$ covariates are non-confounding. The number of covariates P , data size N and cardinality of treatment set K are fixed according to the requirement of experiment and is described in detail in section 5.
- **NEWS:** We consider the publicly available bag-of-words context covariates for NEWS dataset ². The DGP as given in (Schwab et al., 2018) is employed for synthesizing one of multiple treatments and corresponding response for each document (context) in NEWS dataset. This generation process is extended to treatments with dosage levels by (Schwab et al., 2019) and is used for experimental evaluation of continuous valued treatments. The number of covariates P is fixed to 2870 and value for N, K is given in sec. 5 based on experimental requirements.

We use the convention of naming each newly synthesized dataset as a conjunction of the original dataset name and the treatment set cardinality (K) for all experiments in section 5. For example, 'NEWS4' denotes NEWS dataset for $K = 4$ treatment case.

4.2 Metrics

In this subsection, we describe the performance metrics used for evaluating the proposed approach.

4.2.1 PRECISION IN ESTIMATION OF HETEROGENEOUS EFFECT (PEHE)

We use the definition of PEHE as specified in (Schwab et al., 2018) for multiple treatments as:

$$\hat{\epsilon}_P = \frac{1}{\binom{K}{2}} \sum_{m=1}^K \sum_{r=1}^{m-1} \hat{\epsilon}_{P_{m,r}} \quad (15)$$

$$\hat{\epsilon}_{P_{m,r}} = \frac{1}{N} \sum_{n=1}^N ([y_n(m) - y_n(r)] - [\hat{y}_n(m) - \hat{y}_n(r)])^2 \quad (16)$$

2. https://github.com/d909b/perfect_match

where $y_n(m), y_n(r)$ are the response of the n -th individual to treatments T_m and T_r respectively.

4.2.2 MEAN ABSOLUTE PERCENTAGE ERROR (MAPE) OVER AVERAGE TREATMENT EFFECT (ATE)

We use $MAPE_{ATE}$ as a metric to estimate error in predicting average treatment effect for high-cardinality treatments, and is given by:

$$MAPE_{ATE} = \left| \frac{ATE_{actual} - ATE_{pred}}{ATE_{actual}} \right|, \quad (17)$$

where,

$$ATE_{actual} = \frac{1}{N} \sum_{n=1}^N \left(y_n(k) - \frac{1}{K-1} \sum_{l=1, l \neq k}^K y_n(l) \right), \quad (18)$$

and ATE_{pred} is obtained by replacing $y_n(k)$ in the above equation by its predicted value $\hat{y}_n(k)$ for all k .

4.2.3 MEAN INTEGRATED SQUARED ERROR (MISE)

For high cardinality treatments with dosages, we use $MISE$ as a metric (as in (Schwab et al., 2019)). This is the squared error of dosage-response computed across the dosage levels and averaged over all treatments and entire population.

4.2.4 MAPE OVER ATE WITH DOSAGE

We introduce a novel metric $MAPE_{ATE}^{Dos}$ for high cardinality treatments with dosages. This metric is useful for evaluating effect of a dosage level for factual treatment as opposed to counterfactual treatments. It is given by:

$$MAPE_{ATE}^{Dos} = \left| \frac{ATE_{actual}^{Dos} - ATE_{pred}^{Dos}}{ATE_{actual}^{Dos}} \right|, \quad (19)$$

where,

$$ATE_{actual}^{Dos} = \frac{1}{E} \sum_{e=1}^E \left(\frac{1}{N_E} \sum_{n=1}^{N_E} \left(y_n(k_e) - \frac{1}{K-1} \sum_{l=1, l \neq k}^K y_n(l_e) \right) \right), \quad (20)$$

and ATE_{pred}^{Dos} is obtained by replacing $y_n(k_e)$ in the above equation by its predicted value $\hat{y}_n(k_e)$ for all k .

4.3 Baselines

We use the following DNN based approaches to baseline Hi-CI for high cardinality treatments:

- **0-NN**: 0-NN does not account for confounding bias, we thereby bypass the decorrelation network of Hi-CI and \mathbf{X} is directly passed to the outcome network.

- **MultiMBNN**: Matching and balancing based architecture proposed in (Sharma et al., 2020)
- **PM** : Propensity based matching (Schwab et al., 2018) employed for counterfactual regression.
- **Deep-Treat+**: Deep-Treat (Atan et al., 2018) learns bias removing network and policy optimization network independently to learn optimal, personalized treatments from observation data. In order to use Deep-treat as a baseline, we modify Deep-Treat to **Deep-Treat+** and jointly train decorrelation network obtained from Deep-Treat, and outcome network of **Hi-CI** to baseline our approach.
- **DRNet**: is a DNN based technique (Schwab et al., 2019) to infer counterfactual responses when treatments have dosage values. We use this to baseline **Hi-CI** continuous valued treatment case.

5. Experimental Results

We perform extensive experimentation using the proposed **Hi-CI** framework on **Syn** and **NEWS** datasets. Our experimental evaluation is primarily aimed at evaluating the performance of **Hi-CI** under three broad settings: high-cardinality treatments; continuous valued treatments and high number of covariates.

5.1 High-cardinality treatments ($E = 1$)

5.1.1 EFFECT OF INCREASING THE CARDINALITY OF TREATMENT SET

Here, we evaluate **Hi-CI** in scenarios where the cardinality of treatments increases, while $E = 1$. With increase in K , sample size N is also proportionally increased to keep the average number of samples per treatment (given by $\frac{N}{K}$) constant. In Table 1, we report the mean and standard deviation of the performance metrics $PEHE, MAPE_{ATE}$ for **Syn** and **NEWS** datasets. For both the datasets, performance errors increase with increase in K . In the case of **Syn** dataset, error in estimating ATE is much lower than **NEWS** dataset for very large number of treatments. We can explain this by noting that the number of covariates (perhaps confounding too) in **NEWS** dataset are of the order of 2000 whereas in **Syn**, the number of covariates are fixed to 10 with 5 confounding variables.

5.1.2 VARYING NUMBER OF TREATMENTS K FOR FIXED N

We illustrate the performance of the **Hi-CI** framework keeping a sample size of $N = 10000$ while we vary the cardinality of treatment set from $K = 10$ to 100, which implies that we see a decrease in the ratio $\frac{N}{K}$. From Table 2, we observe that for *Syn* dataset, as the average number of samples per treatment decreases, $PEHE$ and $MAPE_{ATE}$ increase. However, for the **NEWS** dataset, no such trend is observed due to a large number but sparse covariates. Furthermore, in Fig. 3 we depict the counterfactual RMSE for **Syn** datasets under this experimental setting. We observe a slight increase in the counterfactual error as K increases, demonstrating that although the problem is harder, **Hi-CI** network prediction performs reasonably well.

Table 1: Effect of increasing K with fixed $\frac{N}{K}$ ratio ($\frac{N}{K} = 285.7$) on high cardinality treatments with $E = 1$. Mean and standard deviation of the metrics for multiple runs with varying seed points is reported.

Dataset	$\sqrt{\hat{\epsilon}_P}$	$MAPE_{ATE}$
Syn35	3.6764, 0.4037	0.074, 0.0188
Syn48	7.4350, 0.1705	0.1494, 0.0048
Syn103	7.0612, 0.5124	0.1681, 0.0054
Syn216	7.7069, 0.1531	0.1943, 0.0101
NEWS35	7.6256, 0.0243	0.393, 0.0095
NEWS48	8.2675, 0.0522	0.4821, 0.0105
NEWS100	8.9334, 0.6425	0.566, 0.0245
NEWS200	9.4679, 0.8524	0.8924, 0.0859

Table 2: Mean and standard deviation of $\sqrt{\hat{\epsilon}_P}$, $MAPE_{ATE}$ for NEWS, Syn datasets with varying $\frac{N}{K}$ ratios. Results are computed for multiple runs of each dataset simulated for varying seed points.

Dataset	$\frac{N}{K}$	$\sqrt{\hat{\epsilon}_P}$	$MAPE_{ATE}$
Syn10	1000	1.6188,0.0262	0.046,0.008
Syn35	285.7	3.6764,0.4037	0.074,0.0188
Syn55	181.8	7.1836,0.8065	0.1378,0.0148
Syn100	100	9.1706,0.8755	0.1812,0.0129
NEWS10	1000	7.8563,0.0214	0.6223,0.115
NEWS35	285.7	7.6256,0.0243	0.393,0.0095
NEWS55	181.8	7.7383,0.0273	0.4515,0.0360
NEWS100	100	8.1432,0.0476	0.507,0.0171

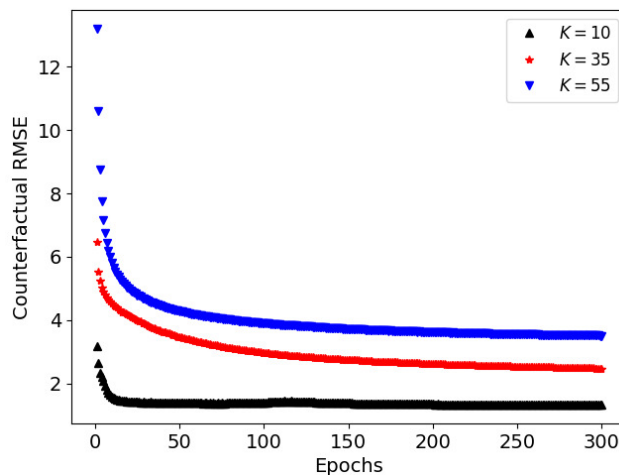


Figure 3: Counterfactual RMSE for increasing K , constant $N = 10000$ on Syn K datasets.

Table 3: Effect of decorrelation loss function $\mathcal{L}_D(\cdot)$ as compared to $\mathcal{L}_{ce}(\cdot) + \mathcal{L}_{ae}(\cdot)$ and $\mathcal{L}_{2,1}(\cdot) + \mathcal{L}_{ae}(\cdot)$ for $E = 1$ case.

Dataset	$\frac{N}{K}$	P	$\sqrt{\hat{\epsilon}_P}$			$MAPE_{ATE}$		
			$\mathcal{L}_1(\cdot)$	$\mathcal{L}_{2,1}(\cdot) + \mathcal{L}_{ae}(\cdot)$	$\mathcal{L}_D(\cdot)$	$\mathcal{L}_1(\cdot)$	$\mathcal{L}_{2,1}(\cdot) + \mathcal{L}_{ae}(\cdot)$	$\mathcal{L}_D(\cdot)$
Syn10	1000	10	1.6390, 0.1125	1.6161, 0.0506	1.6188, 0.0262	0.0645, 0.0243	0.0573, 0.0111	0.046, 0.008
Syn35	285.7	10	5.3784, 0.4538	4.2283, 0.8902	3.6764, 0.4037	0.1686, 0.0181	0.0990, 0.0407	0.074, 0.0188
Syn55	181.8	10	7.5039, 0.4699	7.5173, 0.5540	7.1836, 0.8065	0.1443, 0.0115	0.1472, 0.0103	0.1378, 0.0148
Syn100	100	10	9.7575, 0.7000	11.3353, 0.7624	9.1706, 0.8755	0.2214, 0.0394	0.2138, 0.0067	0.1812, 0.0129
NEWS10	1000	2870	7.8601, 0.0487	7.8541, 0.0285	7.8563, 0.0214	0.6288, 0.0146	0.6325, 0.0027	0.6223, 0.115
NEWS35	285.7	2870	8.3121, 0.0442	8.3425, 0.0600	7.6256, 0.0243	0.4875, 0.0141	0.4874, 0.0081	0.393, 0.0095
NEWS55	181.8	2870	7.8019, 0.0297	7.8212, 0.1648	7.7383, 0.0273	0.4792, 0.0173	0.6454, 0.0945	0.4515, 0.0360
NEWS100	100	2870	8.3275, 0.0792	8.2897, 0.0284	8.1432, 0.0476	0.5028, 0.0169	0.4844, 0.0050	0.507, 0.0171

5.1.3 LOSS FUNCTIONS ANALYSIS

We conducted extensive experimentation to validate the impact of the proposed decorrelation loss function $\mathcal{L}_D(\cdot)$ given in (7), in learning the low-dimensional representation of data as the cardinality of treatments increases. We set the sample size to be constant while K increases, and consequently the ratio $\frac{N}{K}$ decreases. From Table 3, we see that $PEHE$ and $MAPE_{ATE}$ decrease significantly when the lower-dimensional representation is learned using $\mathcal{L}_D(\cdot)$ loss function (7), a combination of losses that caters to reduction in bias via $\mathcal{L}_{ce}(\cdot)$, reduction in information loss via $\mathcal{L}_{ae}(\cdot)$, and similarity-exploiting via $\mathcal{L}_{2,1}(\cdot)$ as compared to case where only $\mathcal{L}_1(\cdot)$ or $\mathcal{L}_{ae}(\cdot) + \mathcal{L}_{2,1}(\cdot)$ is used. Note that $\mathcal{L}_1(\cdot)$ is considered as decorrelation loss in Deep-Treat+.

5.2 Varying number of covariates P

We illustrate the performance of the Hi-CI framework by increasing the number of covariates, retaining the sample size fixed at $N = 10000$, i.e., $\frac{P}{N}$ varies from 0.001 to 0.1. In the context of Syn35 dataset, we see from Table 4 that as the number of covariates increase, $\sqrt{\hat{\epsilon}_P}$ is as low as 3.67 and $MAPE_{ATE}$ is lower than 0.17, thereby showing the strength of the proposed network in handling high dimensional covariates.

 Table 4: Effect of increasing $\frac{P}{N}$ for Syn35 dataset on $PEHE$ ($\sqrt{\hat{\epsilon}_P}$) and error in ATE measured using $MAPE_{ATE}$.

$\frac{P}{N}$	$\sqrt{\hat{\epsilon}_P}$	$MAPE_{ATE}$
0.001	3.6764, 0.4037	0.074, 0.0188
0.005	5.1845, 0.7025	0.1388, 0.0192
0.01	6.2392, 0.3310	0.1557, 0.0132
0.05	6.0466, 0.4325	0.1720, 0.0104
0.1	6.2516, 0.6775	0.1757, 0.0260

5.3 High-cardinality treatments with continuous dosages ($E > 1$)

In Table 5, we illustrate the effect of varying number of dosage levels on the performance metrics for treatments with dosage. Note that the error decreases as the number of dosage levels E increase. We measure the dose-reponse error using $MISE$ and average dosage effect given by $MAPE_{ATE}^{Dos}$ in Table 5 show that varying dosage levels does not impact the performance much. Note that this is partly since context covariates are confounders for treatments, but not for dosage levels in the *NEWS* dataset. Furthermore, In case of synthetic dataset, although covariates are confounders for both treatments and dosages, we see that low-complexity networks are sufficient to capture the dosage-response. As pointed in Sec. 2.1 that our network is designed under the assumption treatment is confounded but not dosage values. However the results for Syn dataset, as seen in Table 5, show that Hi-CI can handle covariates confounding dosages as well.

Table 5: Varying dosage levels E for $N = 10,000$ on treatments with dosage. Mean and standard deviation of \sqrt{MISE} , $MAPE_{ATE}^{Dos}$ computed on test set of datasets generated by varying seeds.

Dataset	E	\sqrt{MISE}	$MAPE_{ATE}^{Dos}$
Syn25	3	2.126, 0.0146	0.1193, 0.0024
Syn25	6	1.980, 0.0157	0.1066, 0.0038
Syn25	8	2.146, 0.014	0.124, 0.0021
Syn25	10	3.148, 0.052	0.162, 0.0046
NEWS25	3	11.2346, 0.1221	0.2462, 0.0584
NEWS25	6	11.4860, 0.1568	0.3254, 0.1221
NEWS25	8	11.0114, 0.0856	0.1457, 0.0462
NEWS25	10	11.9086, 0.2795	0.6890, 0.1258

5.4 Comparative analysis with baselines

In this section, we illustrate the performance of the Hi-CI network as compared to the popular baselines in literature.

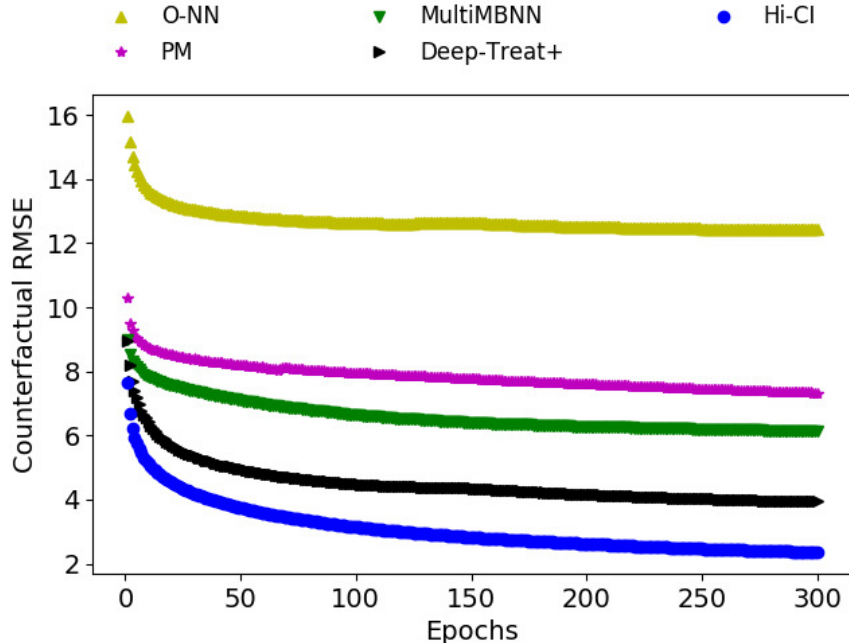
5.4.1 HIGH DIMENSION TREATMENTS AND COVARIATES FOR $E = 1$

In Table 6, we depict the performance of Hi-CI framework as compared to baselines (refer Section 4.3) with varying number of treatments for low and high dimensional covariates. In order to evaluate the performance in high dimensions, NEWS100 with $\frac{P}{N} = 0.287$ is shown to do exceedingly well in terms of both $\sqrt{\hat{\epsilon}_P}$ and $MAPE_{ATE}$, as compared to previous works. It is seen that for lower-cardinality treatment set (Syn4, NEWS4) our approach beats state of art marginally. This is expected behavior since baselines such as (Sharma et al., 2020) and (Schwab et al., 2018) are optimized for such scenarios. However as the number of treatments increase, our approach outperforms baselines by huge margins. This behaviour is observed for both high and low number of covariates.

Table 6: Performance of Hi-CI as compared to state-of-the-art approaches: PM, MultiMBNN for high cardinality treatments.

Dataset	$\frac{P}{N}$	$\frac{N}{K}$	$\sqrt{\epsilon_P}$			$MAPE_{ATE}$		
			PM	MultiMBNN	Hi-CI	PM	MultiMBNN	Hi-CI
Syn4	0.001	2500	1.9004, 0.1124	1.8272, 0.0928	1.3520, 0.0542	0.4249, 0.1142	0.3917, 0.1075	0.0150, 0.0022
Syn10	0.001	1000	0.4249, 0.1142	0.3917, 0.1075	0.0150, 0.0022	5.8976, 0.1175	5.7752, 0.1100	1.6188, 0.0262
Syn35	0.001	285.7	18.5894, 0.2329	17.6520, 0.2032	3.6764, 0.4037	0.4726, 0.0562	0.4528, 0.0864	0.074, 0.0188
Syn100	0.001	100	32.0424, 0.9862	32.304, 0.9652	9.1706, 0.8755	1.1225, 0.1585	1.2854, 0.2012	0.1812, 0.0129
NEWS4	0.287	2500	8.1842, 0.4202	7.6606, 0.4077	6.4120, 0.3016	0.3232, 0.0574	0.1622, 0.0381	0.0984, 0.0245
NEWS10	0.287	1000	9.1540, 0.0245	9.002, 0.0185	7.8563, 0.0214	0.8641, 0.0962	0.7452, 0.105	0.6223, 0.115
NEWS35	0.287	285.7	18.5894, 0.2329	17.6520, 0.2032	3.6764, 0.4037	0.4726, 0.0562	0.4528, 0.0864	0.074, 0.0188
NEWS100	0.287	100	48.3878, 0.5620	49.6386, 0.8520	8.1432, 0.0476	1.9850, 0.1824	2.2014, 0.2350	0.507, 0.0171

In Fig. 4, we depict the counterfactual RMSE obtained using Hi-CI as compared to O-NN, PM, Deep-Treat+, validating our claim of outperforming the state of art approaches for CI.


 Figure 4: Comparison against baselines using counterfactual RMSE metric on Syn10 dataset for $N = 10000$.

5.4.2 HIGH CARDINALITY TREATMENTS WITH CONTINUOUS DOSAGES

In Table 7 we depict the comparative dosage-response values for different datasets averaged over all treatments and individuals, in terms of \sqrt{MISE} . We observe that the Hi-CI framework outperforms the state of the art DNN-based approach, DRNet by a considerable margin for several treatment counts.

Table 7: Comparing with baselines: Hi-CI for continuous treatments, $E > 1$.

Dataset	DRNet	Hi-CI
NEWS2	7.7, 0.2	6.2450, 0.1254
NEWS4	11.5, 0.0	11.0842, 0.1358
NEWS8	10.0, 0.0	8.7540, 0.1032
NEWS16	10.2, 0.0	8.6560, 0.0452

6. Discussions and Conclusions

In CI applications, one commonly encounters situations where there are large number of covariates and large number of treatments in real-world observational studies. The biggest hindrance in such a scenario is in inferring which of the covariates is the actual confounder among the large number of covariates. Furthermore, the complexity of the situation is enhanced since one needs to determine such confounding effects per treatment, for a large number of treatments.

We tackle these seemingly hard scenarios using a generalized Hi-CI framework. Our approach is based on a fundamental assumption that the high-dimensional covariates are often sparse, and can be represented in a low-dimensional space. Naturally, we employed an auto-encoder to represent covariates in a low-dimensional space, without losing much information in the original covariates. Alongside, we also incorporate a decorrelating loss function which ensures that we obtain an equivalent representation of the covariate space with a reduced confounding bias. Furthermore, using the fact that often several treatments/interventions are perhaps similar, we used an embedding in order to obtain a low-dimensional representation of the treatment. In literature, continuous treatments are used, which we addressed by using per-dosage level embedding.

We performed extensive experimental analysis of Hi-CI for synthetic Syn dataset and semi-synthetic NEWS dataset. For high cardinality discrete treatments, we observed that when we increase the number of treatments, not only does Hi-CI perform considerably well in spite of CI being an ill-posed problem, but also is able to outperform the state of art DNN approaches. We further show that the proposed framework handles continuous valued dosages which may or not may be confounded by covariates. We also depicted the decreasing trend in RMSE across epochs, which leads us to understand that the model has indeed trained well. Overall, the theory and results demonstrate that Hi-CI can be regarded as a generalized causal inference framework for tackling high dimension covariates and large number of treatments which may be continuous or discrete valued in observational studies. As a future work, we intend to analyse scenarios where $P \ll N$, which occur in biological applications such as gene intervention studies.

Appendix A.

In this appendix, we provide details of the experiments performed to evaluate Hi-CI framework. Specifically, we describe the dataset preparation, training mechanism and hyperparameter tuning.

A.1 Implementation

A.1.1 ALGORITHM

We provide the implementation details of Hi-CI framework in Algorithm 1 and 2. Specifically, in algorithm 1 we explain the methodology used for splitting dataset \mathbf{D} into train (\mathbf{D}_{CI}), validation (\mathbf{D}_{val}), test (\mathbf{D}_{tst}) sets. We also explain the mechanism for hyperparameter selection. On the otherhand, algorithm 2 outlines the procedure for training Hi-CI for the given set of hyperparameters. We initialise the parameters W of Hi-CI using random normal distribution. We use Adam optimizer with inverse time decay learning rate for gradient descent. In algorithm 1, *hparam_values* specifies the range of hyperparameters for grid-search as in Table 8, *num_unique_treat*(\cdot) returns the number of unique treatments in the dataset passed as argument, *get_gs_hparams*(\cdot) returns set containing exhaustive combination of hyperparameters, *get_best_params*(\cdot) returns Hi-CI parameters corresponding to best validation loss and *get_metric*(\cdot) returns performance metrics of trained Hi-CI on dataset passed as argument. Similarly in algorithm 2, *initialize*(\cdot) initializes parameters of Hi-CI using random normal distribution, *get_random_batches*(\cdot) creates random batches of the dataset with batch size as specified in the argument, *train*(\cdot) trains Hi-CI, *check_convergence*(\cdot) checks for convergence on \mathbf{D}_{val} , *get_final_params*(\cdot) returns learned parameters W_f of Hi-CI and *get_val_loss*(\cdot) returns loss \mathcal{L} on \mathbf{D}_{val} corresponding to W_f .

Algorithm 1 Hi-CI

```

1: procedure HI-CI( $\mathbf{D}$ ,  $K$ , hparam_values,  $E = 1$ )
2:   Split  $\mathbf{D}$  into  $\mathbf{D}_{CI}, \mathbf{D}_{val}, \mathbf{D}_{tst}$ 
3:   while num_unique_treat( $\mathbf{D}_{CI}$ )  $< K$  do
4:     Split  $\mathbf{D}$  into  $\mathbf{D}_{CI}, \mathbf{D}_{val}, \mathbf{D}_{tst}$ 
5:      $\mathcal{L}_{val} = \emptyset, W = \emptyset$ 
6:     gs_hparams = get_gs_hparams(hparam_values)
7:     for gs_hparam in gs_hparams do
8:        $\mathcal{L}_{val}, W \leftarrow \text{trainer}(\text{gs\_hparam}, \mathbf{D}_{CI}, \mathbf{D}_{val})$ 
9:      $W' = \text{get\_best\_params}(\mathcal{L}_{val}, W)$ 
10:     $PEHE, MAPE_{ATE} = \text{get\_metric}(\mathbf{D}_{tst}, W')$ 
    return  $PEHE, MAPE_{ATE}, W'$ 

```

A.1.2 PARAMETER TUNING AND MODEL SELECTION

We select the optimal parameters W' for Hi-CI by performing an exhaustive grid-search on the hyperparameters values mentioned in Table 8.

Algorithm 2 Train_{Hi-CI}

```

1: procedure trainer(gs_hparam,  $\mathbf{D}_{CI}$ ,  $\mathbf{D}_{val}$ )
2:    $W = \text{initialize}()$ 
3:    $total\_epochs = gs\_hparam.total\_epochs$ 
4:    $batch\_size = gs\_hparam.batch\_size$ 
5:   while  $epoch \leq total\_epochs$  do
6:      $\mathbf{D}_{batches} = \text{get\_random\_batches}(\mathbf{D}_{CI}, batch\_size)$ 
7:     for  $\mathbf{D}_{batch}$  in  $\mathbf{D}_{batches}$  do
8:        $W = \text{train}(W, \mathbf{D}_{batch}, gs\_hparam)$ 
9:       if  $\text{check\_convergence}(W, \mathbf{D}_{val})$  then
10:         $W_f = \text{get\_final\_params}(W)$ 
11:        break
12:       $epoch = epoch + 1$ 
13:    $\mathcal{L} = \text{get\_val\_loss}(W_f, \mathbf{D}_{val})$ 
   return  $\mathcal{L}, W_f$ 

```

Table 8: Hyperparameters for grid-search (*hparam_values*)

Parameter	Values
Batch size	64, 128, 256, 512
Total epochs	1000
Learning rate	0.06, 0.08, 0.1, 0.12, 0.14, 0.16
Learning rate decay	0.6, 0.65, 0.7, 0.75
No. of iterations per decay	1, 2
Train set split ratio	0.6
Validation set split ratio	0.2
Test set split ratio	0.2
No. of encoder layers	1, 2, 3, 5, 7
No. of decoder layers	3, 4, 5, 6, 7, 8
No. of outcome layers	3, 4, 5, 6, 7, 8
No. of hidden nodes in encoder layers	100, 150, 200, 250
No. of hidden nodes in decoder layers	100, 175, 250, 325, 400
No. of hidden nodes in outcome network	100, 200, 250, 300, 400, 500
L-2 regularization co-efficient for Φ, Ψ, Ω_e	0.01, 0.001, 0.0001

Table 9: Logistic regression hyperparameters for grid-search

Parameter	Values
Inverse of regularization strength	0.001, 0.01, 0.1, 1, 10
Solver	newton-cg, sag, saga, lbfgs
Tolerance for stopping criteria	1e-4, 1e-2

A.1.3 LEARNING θ_{T_k} :

We use the multi-class logistic regression library of scikit-learn for learning θ_{T_k} in (9). The range of hyperparameters for grid-search in logistic regression is given in Table 9.

A.2 Baselines

In order to make fair comparison of Hi-CI with baselines, we used consistent train, validation and test sets. Additionally, same set of hyperparameters were used for best model selection across all approaches.

A.3 Compute Infrastructure

All experiments in this paper were run on Nvidia Tesla V100 GPU with 32 GB memory and Intel Xeon CPUs with 64 GB RAM.

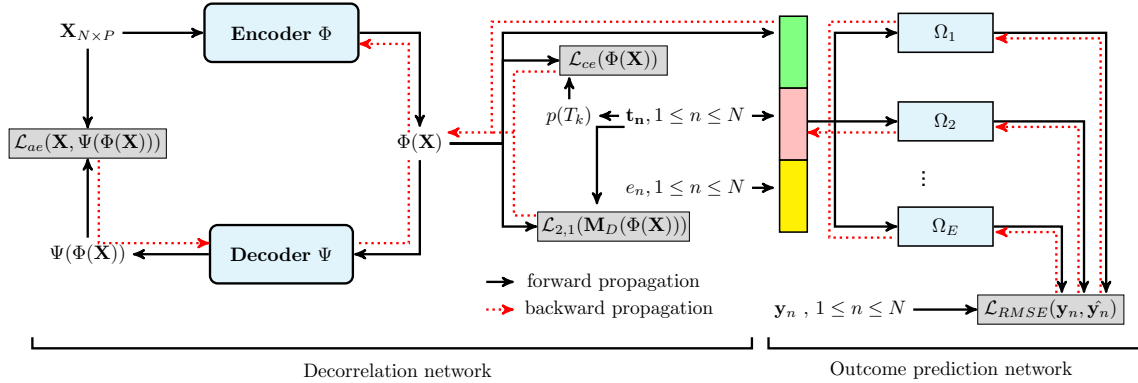


Figure 5: Detailed architecture (with notations) of Hi-CI is depicted. Dotted red arrows highlight joint learning of decorrelating and outcome prediction network. $\mathbf{X} \in \mathbb{R}^{N \times P}$, $\Psi(\Phi(\mathbf{X})) \in \mathbb{R}^{N \times P}$, $\Phi(\mathbf{X}) \in \mathbb{R}^{N \times L}$, $\mathbf{t}_n \in \mathbb{R}^{K \times 1}$, $1 \leq k \leq K$.

References

O. Atan, J. Jordon, and M. van der Schaar. Deep-treat: Learning optimal personalized treatments from observational data using neural networks. In *AAAI*, 2018.

- S. Athey, G. W. Imbens, and S. Wager. Approximate residual balancing: debiased inference of average treatment effects in high dimensions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(4):597–623, 2018.
- A. Belloni, V. Chernozhukov, and C. Hansen. Inference on treatment effects after selection among high-dimensional controls. *The Review of Economic Studies*, 81(2):608–650, 2014.
- L. Bottou, J. Peters, J. Quiñonero-Candela, D. X. Charles, D. M. Chickering, E. Portugaly, D. Ray, P. Simard, and E. Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *JMLR*, 14(1):3207–3260, 2013.
- P. Bühlmann. Causal statistical inference in high dimensions. *Mathematical Methods of Operations Research*, 77(3):357–370, 2013.
- B. Dalessandro, C. Perlich, O. Stitelman, and F. Provost. Causally motivated attribution for online advertising. In *Workshop on Data Mining for Online Advertising and Internet Economy*, pages 1–9, 2012.
- E. Diemert, J. Meynet, P. Galland, and D. Lefortier. Attribution modeling increases efficiency of bidding in display advertising. In *Proceedings of the ADKDD’17*, pages 1–6, 2017.
- J. Fan, K. Imai, H. Liu, Y. Ning, and X. Yang. Improving covariate balancing propensity score: A doubly robust and efficient approach. Technical report, Technical report, Princeton Univ, 2016.
- C. Fong, C. Hazlett, K. Imai, et al. Covariate balancing propensity score for a continuous treatment: Application to the efficacy of political advertisements. *The Annals of Applied Statistics*, 12(1):156–177, 2018.
- D. Galagate. *Causal Inference with a continuous treatment and outcome: Alternative Estimators for parametric dos-response functions with Applications*. PhD thesis, 2016.
- D. Guo, L. Xue, and H. Chen. Cbpps-based inference in nonlinear regression models with missing data. *Open Journal of Statistics*, 6(4):675–684, 2016.
- K. Hirano and G. W. Imbens. The propensity score with continuous treatments. *Applied Bayesian modeling and causal inference from incomplete-data perspectives*, 226164:73–84, 2004.
- D. E. Ho, K. Imai, G. King, and E. A. Stuart. Matching as nonparametric preprocessing for reducing model dependence in parametric causal inference. *Political analysis*, 15(3): 199–236, 2007.
- K. Imai and M. Ratkovic. Covariate balancing propensity score. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76, 2014.
- F. Johansson, U. Shalit, and D. Sontag. Learning representations for counterfactual inference. In *ICML*, pages 3020–3029, 2016.

- E. H. Kennedy, Z. Ma, M. D. McHugh, and D. S. Small. Non-parametric methods for doubly robust estimation of continuous treatment effects. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(4):1229–1245, 2017.
- D. Newman. UCI Machine Learning Repository, 2012. URL <https://archive.ics.uci.edu/ml/datasets/Bag+of+Words>.
- V. Ramachandra. Deep Learning for Causal Inference. *arXiv preprint arXiv:1803.00149*, 2018.
- J. M. Robins, A. Rotnitzky, and L. P. Zhao. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American statistical Association*, 89(427):846–866, 1994.
- P. Schwab, L. Linhardt, and W. Karlen. Perfect Match: A simple method for learning representations for counterfactual inference with neural networks. *arXiv preprint arXiv:1810.00656*, 2018.
- P. Schwab, L. Linhardt, S. Bauer, J. M. Buhmann, and W. Karlen. Learning counterfactual representations for estimating individual dose-response curves. *arXiv preprint arXiv:1902.00981*, 2019.
- U. Shalit, F. D. Johansson, and D. Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *ICML*, volume 70, pages 3076–3085. PMLR, 2017.
- A. Sharma, G. Gupta, R. Prasad, A. Chatterjee, L. Vig, and G. Shroff. MetaCI: Meta-Learning for Causal Inference in a Heterogeneous Population. *NeurIPS CausalML workshop; arXiv preprint arXiv:1912.03960*, 2019.
- A. Sharma, G. Gupta, R. Prasad, A. Chatterjee, L. Vig, and G. Shroff. MultiMBNN: Matched and Balanced Causal Inference with Neural Networks. *ESANN; arXiv preprint arXiv:2004.13446*, 2020.
- W. Sun, P. Wang, D. Yin, J. Yang, and Y. Chang. Causal inference via sparse additive models with application to online advertising. In *AAAI*, 2015.
- J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, J. M. Stuart, C. G. A. R. Network, et al. The cancer genome atlas pan-cancer analysis project. *Nature Genetics*, 45(10):1113, 2013.
- Z. Zhang, Q. Lan, L. Ding, Y. Wang, N. Hassanpour, and R. Greiner. Reducing selection bias in counterfactual reasoning for individual treatment effects estimation. *arXiv preprint arXiv:1912.09040*, 2019.