PDF Download
3652610.pdf
14 January 2026
Total Citations: 1
Total Downloads: 400

Published: 17 May 2024
Online AM: 20 March 2024
Accepted: 03 February 2024
Revised: 15 December 2023
Received: 19 November 2022

Citation in BibTeX format

RESEARCH-ARTICLE

Empowering Predictive Modeling by GAN-based Causal Information Learning

JINWEI ZENG, Beijing National Research Center for Information Science and Technology, Beijing, China

GUOZHEN ZHANG, Beijing National Research Center for Information Science and Technology, Beijing, China

JIAN YUAN, Beijing National Research Center for Information Science and Technology, Beijing, China

YONG LI, Beijing National Research Center for Information Science and Technology, Beijing, China

DEPENG JIN, Beijing National Research Center for Information Science and Technology, Beijing, China

Open Access Support provided by:

Beijing National Research Center for Information Science and Technology

ACM Transactions on Intelligent Systems and Technology, Volume 15, Issue 3 (June 2024)
https://doi.org/10.1145/3652610
EISSN: 2157-6912

Empowering Predictive Modeling by GAN-based Causal Information Learning

JINWEI ZENG, Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Haidian Qu, China
GUOZHEN ZHANG, Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Haidian Qu, China
JIAN YUAN, Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Haidian Qu, China
YONG LI, Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Haidian Qu, China
DEPENG JIN, Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Haidian Qu, China

Generally speaking, we can easily specify many causal relationships in the prediction tasks of ubiquitous computing, such as human activity prediction, mobility prediction, and health prediction. However, most of the existing methods in these fields failed to take advantage of this prior causal knowledge. They typically make predictions only based on correlations in the data, which hinders the prediction performance in real-world scenarios, because a distribution shift between training data and testing data generally exists. To fill in this gap, we proposed a Generative Adversarial Network (GAN)-based Causal Information Learning prediction framework, which can effectively leverage causal information to improve the prediction performance of existing ubiquitous computing deep learning models. Specifically, faced with a unique challenge that the treatment variable, referring to the intervention that influences the target in a causal relationship, is generally continuous in ubiquitous computing, the framework employs a representation learning approach with a GAN-based deep learning model. By projecting all variables except the treatment into a latent space, it effectively minimizes confounding bias and leverages the learned latent representation for accurate predictions. In this way, it deals with the continuous treatment challenge, and in the meantime, it can be easily integrated with existing deep learning models to lift their prediction performance in practical scenarios with causal information. Extensive experiments on two large-scale real-world datasets demonstrate its superior performance over multiple state-of-the-art baselines. We also propose an analytical framework together with extensive experiments to empirically show that our framework achieves better performance gain under two conditions: when the distribution differences between the training data and the testing data are more significant and when the treatment effects are larger. Overall, this work suggests that learning causal information

J. Zeng and G. Zhang contributed equally to this research.

This research has been supported in part by the National Key Research and Development Program of China under Grant No. 2022YFF0606904 and in part by the National Natural Science Foundation of China under Grants No. U21B2036 and No. U20B2060.

Authors' addresses: J. Zeng, G. Zhang, J. Yuan, Y. Li, and D. Jin, Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, 30 Shuangqing Road, Haidian Qu, Beijing Shi, China; e-mail: liyong07@tsinghua.edu.cn.
Author current address: G. Zhang, TsingRoc, 1 Wangzhuang Rd, Haidian Qu, Beijing Shi, China.

ACM 2157-6904/2024/05-ART54
https://doi.org/10.1145/3652610

is a promising way to improve the prediction performance of ubiquitous computing tasks. We open both our dataset and code[1] and call for more research attention in this area.

CCS Concepts: • **Computing methodologies → Neural networks**; *Learning latent representations;*

Additional Key Words and Phrases: Prediction, predictive modeling, causal information learning, GAN

ACM Reference Format:
Jinwei Zeng, Guozhen Zhang, Jian Yuan, Yong Li, and Depeng Jin. 2024. Empowering Predictive Modeling by GAN-based Causal Information Learning. *ACM Trans. Intell. Syst. Technol.* 15, 3, Article 54 (May 2024), 19 pages. https://doi.org/10.1145/3652610

1 INTRODUCTION

Predictive modeling refers to the computational predictions or forecasts about future events or outcomes based on available historical data. With the progress of data collection devices and the boom of data availability, prediction has been a commonly employed technique to inform decision-making. Many researchers have devoted themselves to improving the prediction accuracy of these tasks, because good prediction accuracy is essential for practical applications. However, most of the existing methods make predictions based on correlations, especially the recently developed deep learning ones, which hinders the prediction performance in real-world scenarios. The reason is that a distribution shift between training data and testing data generally exists in practice; thereby the learned correlations between the features and the prediction target can become noise in application [1].

To address this problem, recent works from the computer vision field attempt to distill causal relations from data and utilize them to make predictions [37], because causality means that the data generative process in the training data remains stable in the testing data. In other words, if we focus on the causal features to make predictions, then the prediction performance in practical scenarios could be raised. However, causal information is not fully contained in the observational data [23]. Specifically, there typically does not exist a unique set of causal relationships for given observational data [9]. Furthermore, even if the causal relationships can be determined, we cannot accurately calculate the causal effects due to the general existence of confounding biases [10]. Thus, following these works to infer causal relations from observational data is risky and may in turn introduce noise into prediction.

Generally speaking, human experts can easily specify many causal relationships in prediction tasks in practice. Taking human mobility prediction as an example, according to our common sense, people go out less when it rains. Therefore, the influence of weather on people's travel demand can be regarded as a causal relationship. This characteristic of prediction problems gives us a unique viewpoint: If we can jointly model this informative prior knowledge and the data, then it is possible to boost the prediction performance of existing prediction methods in practical scenarios. Following this lead, this article focuses on building a causal information learning framework to raise the prediction performance of existing deep learning methods in prediction tasks.

Although existing research has made several attempts that enable deep learning models to use causal information to give stable predictions [3, 4, 6, 15], they share two drawbacks that are challenging to solve. First, existing works typically can only incorporate the binary causal information representing whether one intervention has been applied or not. In other words, the treatment variable in these works, which refers to the intervention in the context of causality, is limited to binary

[1] https://github.com/tsinghua-fib-lab/GCIL

form. Since the treatment is binary and has only two possible values, they can define a unique counterfactual and estimate the counterfactual distribution to learn the causal information. However, in predictive modeling, the intensity of interventions may vary, and therefore treatment variables need to be continuous to fully capture such intensity information. Thus, existing methods fail to generalize to this scenario, because we will face infinite counterfactuals if we adopt them. Second, these methods cannot integrate with existing deep learning models and train in an end-to-end manner, which is one of the keys to better performance in most applications.

To deal with the challenge, we proposed a **Generative Adversarial Network (GAN)-based Causal Information Learning (GCIL)** prediction framework, which can effectively leverage given causal relationships to improve the prediction performance of an existing deep learning model in predictive modeling. The main idea is to project all variables except the treatment into a latent space with a minimized confounding bias and use the latent representation and the treatment variable to make predictions. In this way, we avoid the problem of defining infinite counterfactuals. Specifically, we first distinguish the treatment variable from other covariates to avoid the causal information getting lost in the high dimensional latent representation space. Then, to learn causal information to facilitate the prediction performance, we propose to use a generative adversarial network to disentangle the treatment variable from confounders. Finally, we use the disentangled representation together with the treatment to make predictions. The proposed framework can be seamlessly integrated with existing deep learning prediction models and trained end-to-end. We also demonstrate that this framework can easily be extended to multiple treatments by an ensemble method that stacks different models built on a single treatment.

To examine the effectiveness of our proposed method, we conducted extensive experiments on two of the representative prediction tasks, including human activity prediction and travel demand prediction. Experiments on two large-scale real-world datasets on different prediction tasks demonstrate that applying our framework to existing deep learning models can generally improve prediction performance. Further, we propose an analytical framework and conduct extensive experiments on several semi-synthetic datasets to examine the boundary of the framework's effectiveness. The results show that our framework performs better under two conditions: when the distribution differences between the training and testing data are more significant or when the treatment effects are larger.

To sum up, our contributions are as follows:

— To the best of our knowledge, we are the first to propose the causal information learning problem in predictive modeling tasks, and we demonstrate that learning causal information is a promising way to improve the prediction performance of existing predictive modeling models in practical scenarios.

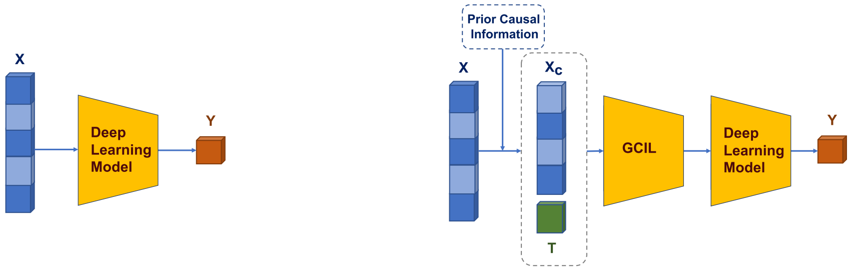
— We propose an effective and generalizable causal information learning framework, GCIL, which can leverage causal information to boost the prediction performance of existing deep learning methods.

— We conduct extensive experiments on two large-scale real-world datasets on different predictive modeling tasks and demonstrate the effectiveness of our framework. We further present an in-depth analysis to empirically examine when it performs better, which provides insights into better model designs.

2 PRELIMINARIES

2.1 Problem Definition

Our goal is to design a causal information learning framework for the general predictive modeling tasks. Specifically, we denote all the input variables as X and the prediction target as Y, which can



(a) The traditional deep learning prediction framework in predictive modeling.

(b) Our proposed prediction framework.

Fig. 1. Comparison between the traditional deep learning prediction framework and our causal information learning framework.

either be a categorical variable or a numeric variable. Our problem can be defined as using $\boldsymbol{X}$ to predict Y with some given prior causal information ξ. In this article, ξ refers to causal relationships given by expert knowledge or common sense, such as the causal relationship between weather and people's mobility. The problem can be formulated as follows:

$$Y = F(\boldsymbol{X}, \xi), \tag{1}$$

where $\boldsymbol{X} \in \mathbb{R}^{n \times N}$. For regression tasks, $Y \in \mathbb{R}^{1 \times N}$, and for classification tasks, $Y \in \{c_1, c_2, \ldots, c_k\}$ where c_i is the ith category. n denotes the dimension of inputs and N denotes the number of instances.

To facilitate methodology modeling, we represent causal information ξ in a mathematical form. We borrow the mathematical representation method of traditional causal inference fields, denoting the intervention that influences the outcome in the causal relationship as the treatment variable $\boldsymbol{T}$ and all other input variables in $\boldsymbol{X}$ as confounding variables $\boldsymbol{X_c}$.

3 METHODOLOGY

3.1 Unified Causal Information Learning Pipeline

As shown in Figure 1(a), traditional deep learning prediction methods in predictive modeling usually adopt an end-to-end framework where all the inputs are fed into a deep learning model to predict the target [17, 35]. Without a specific design, these methods cannot effectively learn causal information from the observational data because of three reasons. First, causal information is not fully contained in the observational data [23]. For a set of observational data, there are usually many possible combinations of causal relationships that could hold to be true, and thus the model is most likely to make predictions relying on some non-causal correlations. Second, the confounding bias, a systematic distortion in the measure of causal effects, generally exists. Therefore, when the model wants to learn causal effects, it could be misled by another factor that affects both the treatment and the prediction targets. Third, existing deep learning methods usually map the input variables into high-dimensional spaces, where the causal signals could easily get lost.

This work aims at utilizing causal information to improve the prediction performance of the above existing predictive modeling models, and thus we first design a unified causal information learning framework that is highly compatible with the traditional deep learning framework, as shown in Figure 1(b). It adds two steps into the traditional prediction pipeline. First, we introduce a causal priori to the model to reformulate the problem, and we denote the causes as the treatment variables $\boldsymbol{T}$, and all other input variables in $\boldsymbol{X}$ as confounding variables $\boldsymbol{X_c}$. In this way, we can reformulate our problem as follows:

$$Y = F(\boldsymbol{X_c}, \boldsymbol{T}), \tag{2}$$

where $\boldsymbol{X_c} \in \mathbb{R}^{(n-n_t)\times N}$, $\boldsymbol{T} \in \mathbb{R}^{n_t \times N}$, and n_t denotes the dimension of treatment variables. In this article, we first show that our proposed framework can handle one-dimensional treatment variables, and then demonstrate how it can easily extend to multi-dimensional treatment variables in Section 3.4. Note that the causal priori specifically refers to the causal relationships given by expert knowledge or our common sense in this article, and such a causal relationship is easy to find in predictive modeling problems. For instance, in terms of human travel demand prediction, temperature, rainfall, and other environmental causes can change one's traveling plan, and thereby temperature, rainfall, and other environmental causes can be regarded as treatments. By distinguishing the treatment variable from other covariates, we avoid the causal information getting lost in the high dimensional latent representation space.

In the second step, we feed the treatments and the confounding variables into a GAN-based causal information learning model, GCIL, to learn causal information from the data to improve the prediction performance. The output of this causal information learning model is an embedding that contains causal information that could be easily used by a deep learning model.

3.2 GAN-based Causal Information Learning

To learn causal information in predictive modeling tasks, we are facing a unique challenge that the treatments are typically continuous variables. Specifically, existing works that attempt to use causal information to give more accurate predictions typically focus on binary treatments [3, 4, 6, 15], such as whether it rains, and they can define counterfactuals and convert the causal information into constraints on factual and counterfactuals accordingly. However, in practical scenarios, such a binary model is oversimplified, which can introduce significant noise. For instance, the travel demand is affected by precipitation. A drizzling has little influence, while a thunderstorm may cause many people to change people's travel plans. Modeling the continuity of the treatment variable is keenly important for improving the prediction performance of predictive modeling tasks, and thus we cannot walk the usual way of defining counterfactuals, because there will be infinite counterfactuals otherwise.

Recent work from the causal inference literature inspired us to propose our GAN-based causal information learning model that can learn causal information for continuous treatments. Although the goal of causal inference is essentially different from ours, the methods of the two tasks can be related, because they both need the model to comprehend causal information. Specifically, researchers from the causal inference literature try to learn causal effects by learning a causal representation in a latent space that balances between the confounding variables' distribution of the treatment group and the control group [14, 28]. Although the treatment group and the control group can only be defined when the treatment is binary, this intuition is valuable for our case.

Following this lead, we propose to map all variables except the treatment into a latent space with a minimized confounding bias and use the latent representation together with the treatment to make predictions, and we designed a GAN-based causal information learning model, GCIL, to achieve this goal. We present its architecture in Figure 2. Since confounding biases originate from the existence of covariates being the cause of both the treatment and the prediction targets, if we can map the confounding variables into a space where the treatment cannot be predicted from the latent space representation, then the confounding biases are minimized. Put differently, the treatments and the confounding variables are disentangled in the latent space. Specifically, we use a generative adversarial network to disentangle treatment from confounders. The goal of the discriminator D is to use the representation output $\boldsymbol{X'}$ by the generator to predict the treatments

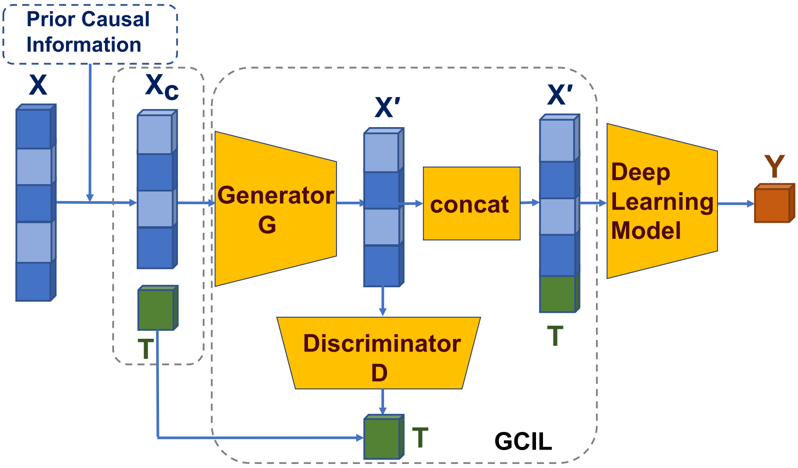


Fig. 2. Architecture of the GCIL framework.

as accurately as possible, while the goal of the generator G is to generate a representation of the confounding variables so that the discriminator cannot use it to predict the treatment. After we obtain the disentangled representation $\boldsymbol{X'}$, we concatenate it with the treatment variables and feed them into the deep learning model to make predictions.

In our implementation, we adopt a two-layer **Multiple-layer Perceptron (MLP)** as the generator G, which is formulated as follows:

$$X' = G(X_c) = \sigma\left(W_g^1\sigma\left(W_g^0X_c + b_g^0\right) + b_g^1\right), \tag{3}$$

where $\boldsymbol{W_g^0} \in \mathbb{R}^{n_{g0}\times(n-1)}$, $\boldsymbol{W_g^1} \in \mathbb{R}^{n_g\times n_{g0}}$, $\boldsymbol{b_g^0} \in \mathbb{R}^{n_{g0}}$, and $\boldsymbol{b_g^1} \in \mathbb{R}^{n_g}$. n represents the dimension of the initial input $\boldsymbol{X}$, n_g represents the dimension of $\boldsymbol{X'}$, and $\sigma(\cdot)$ denotes a non-linear activation function. Without specification, we use **Rectified Linear Unit (ReLU)** in this article. For the discriminator D, we also adopt a two-layer MLP and to predict $\boldsymbol{T}$ the formulation of the prediction $\boldsymbol{D(X')}$ is as follows:

$$D(X') = W_d^1\sigma\left(W_d^0X' + b_d^0\right) + b_d^1, \tag{4}$$

where $\boldsymbol{W_d^0} \in \mathbb{R}^{n_d\times n_g}$,$\boldsymbol{W_d^1} \in \mathbb{R}^{1\times n_d}$, $\boldsymbol{b_d^0} \in \mathbb{R}^{n_d}$ and $\boldsymbol{b_d^1} \in \mathbb{R}^1$.

3.3 Prediction and Training

For the prediction phase, $\boldsymbol{X'}$ is concatenated with $\boldsymbol{T}$ together and inputted into the deep learning model P to predict our target Y. The prediction loss is denoted as $\mathcal{L}_Y$. If Y is a numeric value, then we use the mean absolute error as the prediction loss, which can be formulated as follows:

$$\mathcal{L}_Y(P(\boldsymbol{X'},\boldsymbol{T}),Y) = \frac{1}{N}\sum_{i=1}^{N}|P(\boldsymbol{X'},\boldsymbol{T})^i - Y^i|, \tag{5}$$

where N is the number of instances. If Y is a binary value, then we adopt the cross entropy loss as the prediction loss, which is formulated as follows:

$$\mathcal{L}_Y(P(\boldsymbol{X'},\boldsymbol{T}),Y) = \tfrac{1}{N}\textstyle\sum_{i=1}^{N}(Y^i\log P(\boldsymbol{X'},\boldsymbol{T})^i + (1-Y^i)\log(1-P(\boldsymbol{X'},\boldsymbol{T})^i)). \tag{6}$$

We denote the prediction loss by discriminator D to predict $\boldsymbol{T}$ from $\boldsymbol{X_c}$ as $\mathcal{L}_D$. Since our treatment variable is a continuous numeric variable, we use the mean absolute error as the prediction

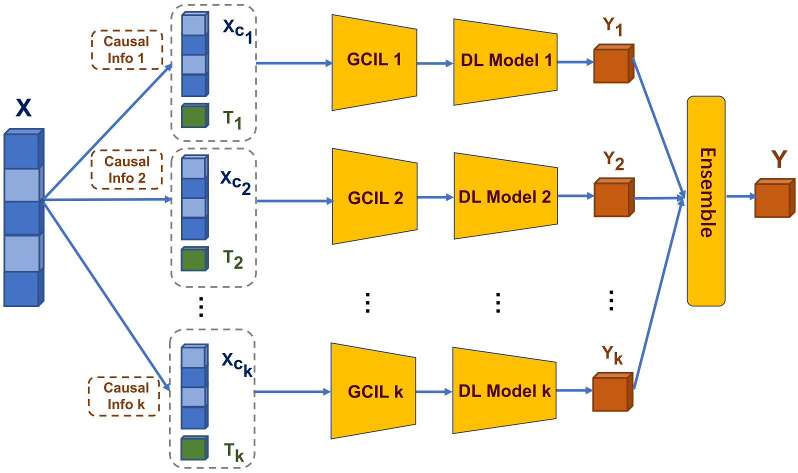


Fig. 3. Our proposed framework when generalized to multiple treatments.

loss, which can be formulated as follows:

$$\mathcal{L}_D(D(\boldsymbol{X'}), \boldsymbol{T}) = \frac{1}{N}\sum_{i=1}^{N} |D(\boldsymbol{X'})^i - \boldsymbol{T}^i|. \tag{7}$$

The goal of the generator G is to maximize $\mathcal{L}_D$ and, in the meantime, minimize the prediction error $\mathcal{L}_Y$, which can be formulated as follows:

$$\min_{G,P}(\mathcal{L}_Y - \beta \times \mathcal{L}_D), \tag{8}$$

where β is a hyper-parameter to balance the two losses. The goal of the discriminator D is to minimize $\mathcal{L}_D$ so that it can predict the treatment as accurately as possible, which is formulated as follows:

$$\min_{D} \mathcal{L}_D. \tag{9}$$

During the adversarial training, we alternate the training of the generator network and the discriminator network after each five iterations.

3.4 Multi-treatment Generalization

Now that we have shown our proposed framework GCIL can leverage one piece of causal information to enhance prediction performance, we can easily extend our framework to deal with more prior causal information, i.e., multi-treatment scenarios, by ensemble learning. The design is presented in Figure 3. Note that this design is only a showcase of the framework's flexibility. More complex methods, such as an attention mechanism, can be used to further improve performance.

We set each prediction head to leverage one piece of prior causal information and ensemble their prediction outcomes to make final predictions. Since each prediction head can leverage its corresponding causal information, combining these prediction heads can help us grasp multiple pieces of causal information, therefore achieving the best performance.

We reformulate our multi-treatment generalization as follows. For the kth prediction head, we formulate the representation $\boldsymbol{X'_k}$ of confounding variables $\boldsymbol{X_{c_k}}$ as

$$\boldsymbol{X'_k} = G_k(\boldsymbol{X_{c_k}}), \tag{10}$$

Table 1. Basic Statistics of Our Three Datasets

Statistics	Beidian	Sharebike	IHDP
Number of instances	20,611	7,260	747
Number of input variables	13	12	26
Prediction type	classification	regression	regression
Dataset type	real-world	real-world	semi-synthesized

where G_k denotes the generator in the GCIL framework for the kth prediction head. Then the deep learning model P_k in this prediction head makes a prediction based on the representation $\boldsymbol{X'_k}$ and the treatment variable $\boldsymbol{T_k}$, which is formulated as follows:

$$\boldsymbol{Y_k} = P_k(\boldsymbol{X'_k}, \boldsymbol{T_k}), \tag{11}$$

where $P_k(\boldsymbol{X'_k}, \boldsymbol{T_k})$ is the prediction of our target by deep learning model P_k.

After training every prediction head, we ensemble the prediction results $\{\boldsymbol{Y_k}, \boldsymbol{k = 1, 2, \ldots, K}\}$ to make the final prediction $\boldsymbol{Y}$. We formulate this process as follows:

$$\boldsymbol{Y} = E(\boldsymbol{Y_1}, \boldsymbol{Y_2}, \ldots, \boldsymbol{Y_k}), \tag{12}$$

where E denotes the ensemble function. For simplicity, we adopt weighted sum as the ensemble function and the weights are hyper-parameters.

4 EVALUATION

To comprehensively evaluate our proposed framework, we take three steps. Specifically, we first use two real-world datasets on the most common predictive modeling tasks to examine the framework's performance. Second, we propose an analytical framework and conduct extensive experiments on several semi-synthetic datasets to investigate when our framework works better. Finally, we examine the performance of our framework when generalizing to the multi-treatment scenario. In this section, we first introduce our experimental setups, including the datasets, baseline methods, and evaluation protocols. Then, we elaborate on the experiment results.

4.1 Experiment Setups

4.1.1 Dataset. To test our framework, we adopt two large-scale real-world datasets of typical prediction scenarios for comprehensive evaluation. We also introduce a semi-synthetic dataset from causal inference benchmarks for further theoretical analysis. The basic statistics of all three datasets are shown in Table 1, and the details of the datasets are as follows:

Sharebike[2]: An open dataset in travel scenario predicting Seoul bike sharing demand. Its covariates include weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), and date information (whether being a holiday, the hour). With all these input variables, we predict the number of public bikes rented at each hour in the Seoul Bike-sharing System. We regard hour as the treatment variable, since peoples' demand to travel is affected by time.

Beidian (Classification): For the human activity prediction scenario, we collect a dataset from the leading social e-commerce platform Beidian in China. On this platform, users can share items with their friends and get rewarded if their friends place an order for the items. The input variables cover users' demographic data (age, gender, city development level, account age), social network data (degree, #already churned neighbours, ...), and behavioral data (total purchase value,

[2] https://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand (Regression)

#purchase, ...). We want to predict whether a user will churn based on these input variables. Considering the role of social connection in users' purchases on this platform, the churn of friends impacts users and may also lead them to churn. Therefore, we regard the number of friends that have churned for every user as the treatment in the prediction.

IHDP[3]: A semi-synthetic dataset from the famous Infant Health and Development Program, which clinically evaluates the efficacy of early intervention in reducing the developmental and health problems of low birth weight premature infants. The original IHDP dataset from the Infant Health and Development Program contains binary 747 observations on 26 covariates, including other features of the infants and their families and evaluation of the pregnancy. Since the initial treatment variable, which is whether the infant receives home visits and attendance at a special child development center, is binary, we synthesize continuous treatment together with outcome referring to Reference [24]. The full equations are

$$\hat{t}|x = 2\frac{x_1}{1+|x_2|} + 2\frac{\max(x_3,x_5,x_6)}{0.2+\min(x_3,x_5,x_6)} + 2\tanh\left(5\frac{\sum_{i\in S_{dis,1}}(x_i-c_1)}{|S_{dis,1}|}\right) - 4 + N(0,0.25), \tag{13}$$

$$t = \frac{1}{1+e^{-\hat{t}}}, \tag{14}$$

$$y|x,t = \frac{\sin(3\pi t)}{1.2-t} \times \tanh\left[5\frac{\sum_{i\in S_{dis,2}}(x_i-c_2)}{|S_{dis,2}|}\right] + \frac{\exp(0.2(x_1-x_6))}{0.5+\min(x_2,x_3,x_5)} + N(0,0.25), \tag{15}$$

where x denotes the input variables, t denotes the continuous treatment, and y denotes the prediction target. $S_{dis,1}=\{x_4,x_7,x_8,x_9,x_{10},x_{11},x_{13},x_{14}\}$, $S_{dis,2}=\{x_{16},x_{17},x_{18},x_{19},x_{20},x_{22},x_{24},x_{25}\}$, $c_1 = \text{mean}(\frac{\sum_{i\in S_{dis,1}} x_i}{|S_{dis,1}|})$, and $c_2 = \text{mean}(\frac{\sum_{i\in S_{dis,2}} x_i}{|S_{dis,2}|})$. Here, mean$(\cdot)$ represents calculating the average.

4.1.2 Baselines. To evaluate the performance of our model, we compare our framework with eight prediction methods from two groups. One group contains some well-established deep learning models that are generally used across different predictive modeling tasks. We set out to examine whether there are performance increases after applying our proposed framework to these baselines. The other group contains some state-of-the-art models that utilize causal information for prediction.

Well-established Non-causal Deep Learning Models:

- MLP [8]: A classical neural network that is composed of multiple layers of perceptrons with threshold activation. In our experiment, we adopt a three-layer MLP. For regression tasks, all threshold activation adopts ReLU activation. For classification tasks, the threshold activation on the last layer uses Sigmoid activation.
- ResMLP [32]: It is a deep residual neural network for the regression/classification of nonlinear functions. It typically skips connections, or shortcuts to jump over some layers. Each layer is a fully connected layer. For a fair comparison, our ResMLP network consists of three layers. To fit in the regression/classification task, the threshold activation on the last layer is set as ReLU/Sigmoid.
- DenseMLP [12]: It is an adaption of the well-established **Densely Connected Convolutional Network (DenseNet)**, which was originally designed for image processing to non-image prediction. The novel DenseMLP model replaced the original convolution and pooling layers with fully connected layers. The original concatenation shortcuts are maintained to reuse the feature. We also set the regression/classification network to be of

[3] https://www.icpsr.umich.edu/web/HMCA/studies/9795

three layers. The threshold activation on the last layer can be ReLU or Sigmoid depending on the prediction type.

To notice, these general predictive baselines we are comparing with are also widely used and are state-of-the-art methods in the specified domains of our evaluation. In the domain of churn prediction, MLP has been widely used to capture the correlations between different attributes and users' intention of churn [11, 31]. In the domain of sharebike demand modeling, Zhang et al. [39] adopt the multi-layer perceptron to forecast the bicycle rental demand and so on. Therefore, the superior performance of our framework over these baselines validates the applicability and contribution of our model to these specific fields.

Causal Prediction Models: There are several existing methods that incorporate causal learning into prediction. The former three models are originally designed to predict the causal effect of a given treatment variable. We adapt these frameworks to predict a target given a causal relationship between the treatment variable and the target. The latter two models attempt to discard the spurious correlations among inputs and learn the stable structure underlying the dataset to improve prediction stability.

- Tarnet [28]: This network splits the continuous treatment variable into several intervals and trains a deep learning prediction model for every interval to predict causal effects, respectively. For each deep learning prediction model, the inputs are the confounding variables of the instances whose treatment variable lies in this treatment interval. To adapt this model to our task, we train every deep learning model to predict a target. For a fair comparison, we set every deep learning prediction model to be of three layers.
- DRNet [27]: This network also discretizes the treatment and trains a prediction head for every treatment interval. Not only the confounding variables but the treatment variable are inputted together into the corresponding prediction head for prediction. For adaptation, we also make every deep learning model predict a target. For a fair comparison, we set every prediction head as a deep learning model of three layers.
- VCNet [24]: Instead of building multiple treatment heads, this network adopts a varying coefficient neural network to predict the causal effect. The varying coefficients are formulated by the continuous treatment variable. For adaption, we also set the neural network to predict a target. For a fair comparison, we use a truncated polynomial basis with degree 2 so that the number of parameters of VCNet is equal to that of Tarnet and DRNet.
- SRDO [30]: Considering that the collinearity among input variables will lead to instability of prediction results, the authors proposed a sample reweighted decorrelation operator to pre-calculate the sample weights, which are then added to the training.
- StableNet [37] This method attempts to learn the sample weights and train the prediction model simultaneously by iteratively optimizing the decorrelation loss and training loss.

4.2 Evaluation Metrics

In the evaluation, we perform five-fold cross-validation on our framework. For the performance evaluation, since there are two forms of prediction: classification and regression, we design two sets of evaluation protocols, respectively.

For classification evaluation: We adopt F1-score and **Area Under the Receiver Operating Characteristic Curve (AUC)** as metrics, which are commonly adopted in classification problems.

F1-score: It combines two important metrics, precision and recall, into a single metric by taking their harmonic mean:

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall}. \tag{16}$$

AUC: This is a metric to calculate the possibility that the classifier ranks a randomly chosen positive instance higher than a randomly chosen negative one, which can be expressed as follows:

$$AUC = \frac{\sum_{i \in positiveclass} rank_i - \frac{M(M+1)}{2}}{M \times N}, \tag{17}$$

where $rank_i$ denotes the rank of instance i in the prediction, M denotes the total number of positive instances and N denotes the total number of negative instances.

For regression evaluation: We adopt MAE and sMAPE as evaluation metrics. The former measures the absolute error and the latter measures the relative error.

MAE: Mean absolute error, which can be formulated as follows:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |\hat{y_i} - y_i|, \tag{18}$$

where N denotes the total number of predicted instances, $\hat{y_i}$ denotes the prediction of instance i and y_i denotes the true value of instance i.

sMAPE: symmetric Mean Absolute Percentage Error, which can be expressed as follows:

$$sMAPE = \frac{1}{N} \sum_{i=1}^{N} \frac{|\hat{y_i} - y_i|}{(|\hat{y_i}| + |y_i|)/2}, \tag{19}$$

where N denotes the total number of predicted instances, $\hat{y_i}$ denotes the prediction of instance i and y_i denotes the true value of instance i.

4.3 Parameter Settings and Implementation Details

During the training process of our proposed framework, we iteratively train the generator G and discriminator D. Every five epochs' training of the generator G and deep learning model P is followed by five epochs' training of the discriminator D. In our experiment, without specification, we use the ReLU function as the activation function. We use the Adam Optimizer [16] for the gradient-based model optimization in a mini-batch mode, and we perform a grid search on the learning rates, batch size, the trade-off weight β to find the optimal hyper-parameters. Specifically, we search the learning rates $\in [1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2, 5e-2, 1e-1]$, batch sizes $\in \{32, 64, 128, 256\}$, and the trade-off weight $\beta \in (0, 1)$. For all baselines, we also tune their learning rates in the range of $[1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2, 5e-2, 1e-1]$. For MLP, ResMLP, and DenseMLP and their GCIL variants, their prediction head consists of three layers, respectively, for a fair comparison. The dimensions of the hidden layers are set as 32. For Tarnet and DRnet, we set the block number as 5 to divide the continuous treatment into 5 blocks. For VCNet, consistent with its setting stated in the article, we use a truncated polynomial basis with degree 2 and two knots at 1/3, 2/3, so it also has 5 bases. For SRDO, the hyperparameters are the learning rate and the batch size, which are also tuned in the mentioned range. For StableNet, apart from the learning rate and the batch size, we also tune the smoothing parameter α, which balances the long-term and short-term memories during representation update, in the range of 0.2, 0.4, 0.6, 0.8. We use early stopping to obtain the model with the best performance.

4.4 Overall Performance

To examine the effectiveness of our framework, we evaluate three variants with different choices of deep learning models, including MLP [8], ResMLP [32], and DenseMLP [12]. The performance of our frameworks and baselines are shown in Table 2, from which we derive three key observations and several insights.

Table 2. Performance Evaluation Results on the Beidian and the Sharebike Datasets

	Beidian (Classification)		Sharebike (Regression)	
Models	F1-score	AUC	MAE	sMAPE
MLP [8]	0.6599 ± 0.0075	0.6948 ± 0.0034	0.0553 ± 0.0016	0.4544 ± 0.0153
ResMLP [32]	0.6634 ± 0.0066	0.6956 ± 0.0030	0.0441 ± 0.0013	0.3796 ± 0.0220
DenseMLP [12]	0.6563 ± 0.0076	0.6943 ± 0.0047	0.0454 ± 0.0014	0.3830 ± 0.0182
Tarnet [28]	0.5581 ± 0.0039	0.5247 ± 0.0026	0.1221 ± 0.0002	0.7357 ± 0.0005
DRNet [27]	0.5300 ± 0.0461	0.5431 ± 0.0354	0.1167 ± 0.0004	0.6921 ± 0.0013
VCNet [24]	0.5958 ± 0.0326	0.5757 ± 0.0458	0.1196 ± 0.0026	0.7028 ± 0.0234
SRDO [30]	0.6574 ± 0.0076	0.6852 ± 0.0038	0.0569 ± 0.0026	0.5003 ± 0.0233
StableNet [37]	0.6568 ± 0.0095	0.6910 ± 0.0053	0.0594 ± 0.0031	0.4789 ± 0.0146
MLP+GCIL	0.6730 ± 0.0012$^+$	0.7006 ± 0.0006$^+$	0.0460 ± 0.0009	0.3980 ± 0.0101
ResMLP+GCIL	0.6726 ± 0.0015$^+$	0.7009 ± 0.0024$^+$	**0.0417 ± 0.0003$^+$**	**0.3516 ± 0.0095$^+$**
DenseMLP+GCIL	**0.6732 ± 0.0015$^+$**	**0.7011 ± 0.0011$^+$**	0.0424 ± 0.0015$^+$	0.3721 ± 0.0226$^+$

"+" indicates the improvement of GCIL over the best baseline is significant at the level of 0.05 in the Wilcoxon signed-rank test.

- **GCIL's consistent improvement over various deep learning models.** As we can see from Table 2, applying our GCIL framework to MLP, ResMLP, and DenseMLP all bring significant performance gain, which validates the effectiveness of our proposed framework. Specifically, for the Beidian dataset, the gains are 0.86%, 0.53%, and 1.04%. For the Sharebike dataset, the performance gains are 20.22%, 5.44%, and 6.61%. All improvements are significant at the level of 0.05. The consistent performance gains also suggest that learning causal information to improve the prediction performance of deep learning models is a promising way.
- **GCIL's superiority over SOTA causal information learning methods.** All models with the GCIL framework outperform state-of-the-art causal information learning methods across different prediction tasks. Compared with the best baseline method, the performance gain of the F1-score for the Beidian dataset is 2.40%. The MAE improvement for the Sharebike dataset is 26.7%. Specifically, our framework greatly outperforms Tarnet [28] and DRNet [27], which may originate from the information loss in discretizing the continuous treatment into several intervals. Although VCNet [24] shows a relatively higher performance, it lacks a clear deconfounding mechanism, rendering it susceptible to confounding bias when utilizing causal information. SRDO [30] and StableNet [37] both employ a reweighting mechanism to decorrelate covariates from treatment. However, reweighting only achieves partial decorrelation, whereas mapping covariates into a new representation space enables complete decorrelation. Thus, we can conclude that our framework leverages causal information better than existing state-of-the-art methods, thereby exhibiting the greatest performance.
- **GCIL's robustness over different scenarios.** The proposed GCIL framework brings performance gain across both regression and classification tasks. Such consistent improvement demonstrates the robustness of the framework. Specifically, the average performance gain of the F1-score by applying our framework to the three deep learning models is 1.99% for the Beidian dataset, while the average performance gain of MAE is 10.76% for the Sharebike dataset.

4.5 When Does Our Framework Work Better?

As shown in Table 2, although there are consistent improvements by integrating GCIL with deep learning models for all real-world datasets, the performance gain brought by GCIL varies. Thus, an important question comes to the fore: when will our framework work, and when does it work

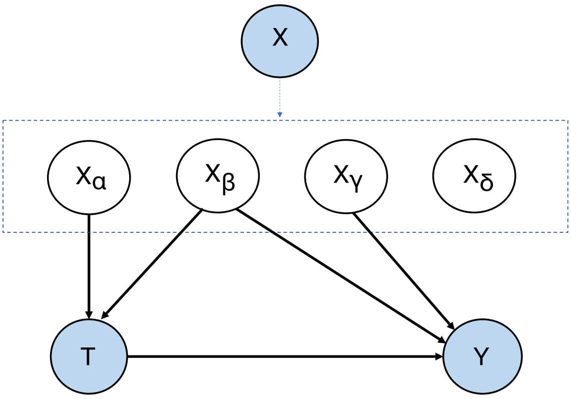


Fig. 4. General decomposition of a given set of confounding variables.

better? To answer this question, we propose an analytical framework based on a causal graph and then conduct extensive experiments to provide empirical evidence.

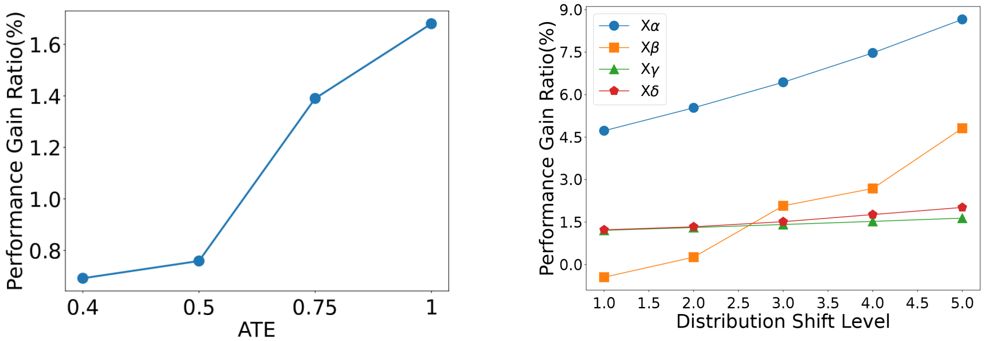
4.5.1 Analytical Framework. In general, the causal relationships among the treatment variable $\boldsymbol{T}$, the confounding variables $\boldsymbol{X}$, and the prediction target $\boldsymbol{Y}$ can be modeled by the causal graph in Figure 4, where the confounding variables can be decomposed into four sub-variables, including $\boldsymbol{X_\alpha}, \boldsymbol{X_\beta}, \boldsymbol{X_\gamma}$, and $\boldsymbol{X_\delta}$. $\boldsymbol{X_\alpha}$ is the cause of $\boldsymbol{T}$. $\boldsymbol{X_\gamma}$ is the cause of $\boldsymbol{Y}$. $\boldsymbol{X_\beta}$ is the cause of both $\boldsymbol{T}$ and $\boldsymbol{Y}$, and $\boldsymbol{X_\delta}$ has no causal relationship with both $\boldsymbol{T}$ or $\boldsymbol{Y}$. According to this framework, our framework's disentangling $\boldsymbol{X}$ and $\boldsymbol{T}$ is equal to removing both the causal arrow pointing from $\boldsymbol{X_\alpha}$ and $\boldsymbol{X_\beta}$ to $\boldsymbol{T}$. Then, our framework makes predictions focusing on the learned causal information between $\boldsymbol{T}$ and $\boldsymbol{Y}$. Thus, there are two most important factors that affect the prediction performance of the proposed framework. One is the scale of the average treatment effect of $\boldsymbol{T}$ on $\boldsymbol{Y}$. The other is the distribution shift of the confounding variables between the training set and the testing set, which is also referred to as the **out-of-distribution (OOD)** problem [37].

4.5.2 Experiment Results. Following this lead, we synthesize several datasets based on the well-known IHDP dataset [24], which clinically evaluates the efficacy of early intervention in reducing the developmental and health problems of low-birth-weight premature infants. By experimenting on datasets synthesized with different treatment effects and distribution shift extent, we examine the impact of the two factors on the performance gains of the framework.

First, we examine the correlation between the scale of the average treatment effects and the performance gain by manipulating the generation process of the synthetic IHDP dataset. Specifically, we derive datasets with different scales of causal effects by adding a coefficient A on the terms that contain the treatment variable in Equation (20), which can be formulated as follows:

$$y|x,t = A \times \frac{\sin(3\pi t)}{1.2-t} \times \tanh\left[5\frac{\sum_{i\in S_{dis,2}}(x_i - c_2)}{|S_{dis,2}|}\right] + \frac{\exp(0.2(x_1 - x_6))}{0.5 + \min(x_2, x_3, x_5)} + N(0, 0.25). \tag{20}$$

By changing A, we change the causal effect of the treatment variable T on Y by A times. We examine the performance gain of our proposed framework on the datasets generated with an A of 0.4, 0.5, 0.75, and 1, and the results are shown in Figure 5(a). As we can see, as A increases, i.e., the causal effect of the given relationship gets larger, the performance gain by our framework increases as well. In other words, our framework performs better when the average treatment effect is larger. This experiment gives an empirical explanation for the results that the performance gain of the proposed framework differs in different scenarios.



(a) The relationship between the average treatment effect (ATE) and the performance gain brought by our proposed framework.

(b) The relationship between the distribution shift level of four types of variables and the performance gain brought by our proposed framework.

Fig. 5. Experimental results on when our framework works better.

Second, we examine the influence of the distribution shift of the confounding variables between the training data and the testing data on the performance gain of our framework. To achieve this goal, we first define an OOD score D to measure the distribution differences and then derive datasets with different OOD scores of $\boldsymbol{X_\alpha}$, $\boldsymbol{X_\beta}$, $\boldsymbol{X_\gamma}$, and $\boldsymbol{X_\delta}$, respectively, so that we can obtain a comprehensive understanding of the framework under OOD conditions. Specifically, given a set of variables, we define the OOD score as the average KL distance between their distributions in the training data and the testing data, which can be formulated as follows:

$$d_{KL}^{i} = D_{KL}(x_{train,i}, x_{test,i}) = \sum_{i=1}^{n} p_{train}(x_i) log \frac{p_{train}(x_i)}{p_{test}(x_i)}, \tag{21}$$

$$D = \frac{1}{N} \sum_{i \in X} d_{KL}^{i}, \tag{22}$$

where D_{KL} denotes calculating KL divergence, and N denotes number of variables in X.

To derive datasets with different distribution shift levels, we first simplified the generation process of the IHDP dataset to ten confounding variables so that we can easily distinguish $\boldsymbol{X_\alpha}, \boldsymbol{X_\beta}, \boldsymbol{X_\gamma}$, and $\boldsymbol{X_\delta}$ and control the distribution shift of them, respectively. The simplified generation process of $\boldsymbol{T}$ and $\boldsymbol{Y}$ are as follows:

$$t|x = \frac{2x_1}{1+x_2} + 2\frac{\max(x_3, x_5)}{0.2 + \min(x_3, x_5)} + 2 \times \tanh(5x_9) + N(0, 0.25), \tag{23}$$

$$y|x, t = \frac{\sin(3\pi t)}{t} \tanh[5(x_6 + x_{10})] + 10 \times \frac{\exp(0.2(x_1 - x_6))}{0.5 + 5\min(x_2, x_3, x_5)} + N(0, 0.02), \tag{24}$$

where $\boldsymbol{X_\alpha} = \{x_9\}$, $\boldsymbol{X_\beta} = \{x_1, x_2, x_3, x_5\}$, $\boldsymbol{X_\gamma} = \{x_6, x_{10}\}$, and $\boldsymbol{X_\delta} = \{x_4, x_7, x_8\}$, respectively. Then, we fix the training data and manipulate the testing data by sampling a specific set of variables from normal distributions that have different mean with that of the training data for $\boldsymbol{X_\alpha}$, $\boldsymbol{X_\beta}$, $\boldsymbol{X_\gamma}$, and $\boldsymbol{X_\delta}$, respectively. In this way, we get four sets of datasets with different distribution shift levels of a specific set of confounding variables having other variables controlled.

We test our framework on each set of these datasets and show the results in Figure 5, where we can draw two key insights. First, our framework brings consistent performance gain in all OOD scenarios. In other words, our framework is valuable when there exists a distribution shift between the training data and the testing data, which is a common case in predictive modeling. Second, for

Table 3. Performance Comparison Between the Ensemble Model and Two Base Models

	Treatment	MAE	sMAPE
Submodel A	Hour	0.0460	0.3980
Submodel B	Precipitation	0.0421	0.3669
Ensemble Model	Both	0.0415	0.3487

all types of confounding variables, including X_α, X_β, X_γ, and X_δ, when the distribution shift level increases, the performance gain of applying our framework becomes larger, which demonstrates that our framework can effectively model causal information.

To sum up, our framework achieves better performance gains under two conditions: when the treatment effects are larger or when the distribution differences between the training data and the testing data are more significant.

4.6 Multi-treatment Generalization Evaluation

After we present the GCIL framework that deals with a single causal relationship, we demonstrate that it can be easily extended to multiple treatments by ensembling GCIL models built on different treatment variables together in this section. The ensemble framework is depicted in Figure 3. Without loss of generality, we use the Sharebike dataset to showcase its effectiveness. Specifically, we adopt weighted sum as the ensemble function and MLP as the deep learning model. We ensemble two basic models with the hour and the precipitation as the treatment variable and denote them as Submodel A and Submodel B, respectively. After training both two models, we calculate the weighted sum of the two outputs as the final prediction results. Specifically, we restrict the sum of the two weights to equal 1 and iterate the weight of Submodel A from 0.01 to 0.99 with an interval of 0.01. The best ensemble result, for which the weight for Submodel A is 0.27, can lift the model performance significantly. Specifically, as listed in Table 3, our ensemble model outperforms the best base models by 1.43% and 9.78% in terms of MAE and sMAPE, respectively, which suggests the generalizability of our proposed method. We visualize the performance of the ensemble model with varying weights in Figure 6. As we can see, the ensemble model with any weight can outperform the original Submodel B. A remarkable observation from our investigation reveals that an approximate majority, approximately 60 percent, of the weights corresponded to the ensemble model demonstrating superior performance. All these findings underscore the efficacy of our model in leveraging multiple causal information.

5 RELATED WORK AND DISCUSSION

5.1 Prediction by Correlations

As massive data sources become available, predictive modeling has been a powerful tool for decision-making informing [2, 22]. The general prediction tasks, which involve mining the pattern and correlations between attributes and thereby inferring unknown situations, have long been an important research problem. With the progress of deep learning, MLP [8] has been proposed as a backbone for predictive modeling. Then Resnet architecture [32] and DenseNet [12] architecture have been further proposed to prevent gradient vanishing and gradient explosion and thereby boost the predictive performance. Based on these general prediction models, various prediction models targeting specific fields are proposed, including human activity prediction [17, 18, 36], mobility prediction [7, 13], health prediction [20, 38], and travel demand prediction [26]. For example, Krishna et al. [17] proposed an Long Short-term Memory-based deep learning model to predict human activity given their past behaviors. Kwon et al. [18] predicts the churn of users leveraging

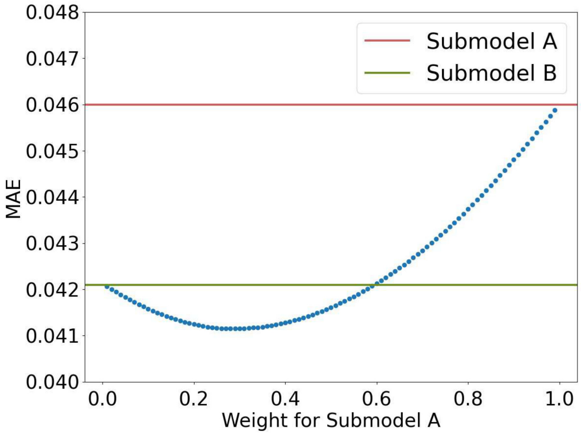


Fig. 6. Performance results of our Ensemble model with varying weights.

venue-specific features. Jiang et al. [13] designed a region of interest modeling approach to predict urban human mobility. Lin et al. [20] develop an interpretable machine learning model to predict individual health conditions with mobility data.

In the aforementioned tasks, existing models typically make predictions based on correlations, which hinders their performance in practical scenarios. In this article, we suggest that their prediction performance in practical scenarios can be further improved by modeling causal information. The reason lies in three aspects. First, it is common in predictive modeling that the training data distribution deviates from the testing data. In this case, correlation-based models trained on training data may depend on correlations that do not hold in the testing data and thereby leading to poor performance. Second, the causal information is not fully contained in the observational data [23]. Thus, guiding the model to integrate causal relationships to make predictions can improve the model performance by an information gain. Third, without a specific design, existing models cannot effectively learn causal information from the observational data due to the general existence of confounding bias.

5.2 Prediction by Causality

5.2.1 Learning Causal Information for Deep Learning Models. In recent studies, researchers have begun exploring the integration of causal information into deep learning models to enhance their performance [3, 19, 21, 29, 40]. Existing research in this area can be broadly categorized into two main approaches. The first category aims to leverage causal relationships to mitigate bias in observed data [3, 40]. The second category focuses on learning causal effects to improve prediction stability [19, 29, 30, 37]. Our work is closely aligned with the latter one.

Targeted at the field of stable learning [5], Shen et al. [30, 37] addressed the out-of-distribution hazards of existing predictive methods by adopting covariate balancing strategies in causal inference to minimize the distribution shifts between different datasets and increase the stability of prediction. However, these methods fail to utilize explicitly given causal information. Some other methods learn causal effects by estimating the counterfactual data distribution, and then they use the estimated causal effects as an additional input of the predictor. For example, Li et al. [19] use a traditional propensity score matching method to estimate the causal effects and use them as a feature of the prediction network. However, a significant challenge for these methods to adapt to predictive modeling scenarios is that they can only handle binary treatment. In predictive modeling, treatment variables are generally continuous. Thus, existing methods fail to generalize to this scenario, because we will face infinite counterfactuals if we adopt them. This work proposes a causal information learning framework that can effectively deal with continuous treatment variables.

5.2.2 Causal Inference with Deep Learning. Causal inference has been a long-standing problem in research history [25]. Recently, researchers begin to use deep learning methods to facilitate this task. Although the goal of causal inference is essentially different from our tasks', the methods of the two tasks are related. Specifically, causal inference focuses on determining the effects of a treatment variable on the outcome variable [33], while this work cares more about the prediction accuracy of the outcome variable. The subtle relationship between the two tasks is that they both want the model to learn from causal information, and thereby the intuition behind the model design of the two tasks can be similar.

Overall, there are mainly two types of deep learning methods for causal inference. One is learning causal representations [14, 28]. The other is counterfactual prediction [34, 41]. However, most of the existing works are designed for binary treatments and are hard to generalize to continuous ones. Some recent efforts address this problem by estimating the average dose-response function rather than the treatment effect. Schwab et al. [27] propose DRNet to discretize the treatment and train different prediction networks for different treatments. Based on DRNet, Nie et al. [24] further take the continuity of the dose-response curve into consideration and use a varying coefficient model to estimate the dose-response curve. In this work, we follow the research line of causal representation learning and propose a GAN-based causal information learning framework for continuous treatment to improve the prediction performance of predictive modeling tasks, which consistently outperforms the above approaches across different prediction tasks.

6 CONCLUSION AND FUTURE WORK

In this article, we propose the GCIL prediction framework that fully incorporates given causal information into the prediction process of deep learning models and improves their performance. Our work bridges the gap between the general prediction methods nowadays and their limited utilization of existing abundant causal information and knowledge. By projecting all variables except the treatment into a latent space decorrelated to the treatment, our work is capable of dealing with causal information, even when the intervention is continuous. In conclusion, the highlight of our work lies in the full utilization of general causal information to enhance prediction accuracy.

We evaluate our framework on two real-world datasets, which shows its effectiveness and robustness compared with the state-of-the-art baselines. We further proposed an analytical framework and conducted extensive experiments to show that our framework can achieve better performance gains under two conditions: when the distribution differences between the training data and the testing data are more significant or when the treatment effects are larger. Overall, this work suggests that learning causal information is a promising way to improve the prediction performance of predictive modeling tasks. A meaningful future direction is to explore how to jointly utilize several causal information to improve the prediction performance better. In addition, it is also valuable to explore how to apply our framework to dynamic scenarios where temporal causality relationships exist.

REFERENCES

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. Retrieved from https://arXiv:1907.02893

[2] Jorge Luis Victória Barbosa. 2015. Ubiquitous computing: Applications and research opportunities. In *Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research (ICCIC'15)*. IEEE, 1–8.

[3] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems.* 104–112.

[4] Bo Chen, Xiaoshu Sun, Decai Li, Yuqing He, and Chunsheng Hua. 2021. SCR-Graph: Spatial-causal relationships-based graph reasoning network for human action prediction. In *Proceedings of the 2nd International Conference on Computing and Data Science.* 1–9.

[5] Peng Cui and Susan Athey. 2022. Stable learning establishes some common ground between causal inference and machine learning. *Nature Mach. Intell.* 4, 2 (2022), 110–115.

[6] Barbra A. Dickerman and Miguel A. Hernán. 2020. Counterfactual prediction is not only for causal inference. *Eur. J. Epidemiol.* 35, 7 (2020), 615–617.

[7] Jie Feng, Can Rong, Funing Sun, Diansheng Guo, and Yong Li. 2020. PMF: A privacy-preserving human mobility prediction framework via federated learning. *Proc. ACM Interact., Mobile, Wear. Ubiq. Technol.* 4, 1 (2020), 1–21.

[8] Matt W. Gardner and S. R. Dorling. 1998. Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences. *Atmos. Environ.* 32, 14–15 (1998), 2627–2636.

[9] Clark Glymour, Kun Zhang, and Peter Spirtes. 2019. Review of causal discovery methods based on graphical models. *Front. Genet.* 10 (2019), 524.

[10] Sander Greenland, Judea Pearl, and James M. Robins. 1999. Confounding and collapsibility in causal inference. *Stat. Sci.* 14, 1 (1999), 29–46.

[11] Mohammad Ridwan Ismail, Mohd Khalid Awang, M. Nordin A. Rahman, and Mokhairi Makhtar. 2015. A multi-layer perceptron approach for customer churn prediction. *Int. J. Multimedia Ubiq. Eng.* 10, 7 (2015), 213–222.

[12] Chao Jiang, Canchen Jiang, Dongwei Chen, and Fei Hu. 2021. Densely connected neural networks for nonlinear regression. Retrieved from https://arXiv:2108.00864

[13] Renhe Jiang, Xuan Song, Zipei Fan, Tianqi Xia, Quanjun Chen, Qi Chen, and Ryosuke Shibasaki. 2018. Deep ROI-based modeling for urban human mobility prediction. *Proc. ACM Interact., Mobile, Wear. Ubiq. Technol.* 2, 1 (2018), 1–29.

[14] Fredrik Johansson, Uri Shalit, and David Sontag. 2016. Learning representations for counterfactual inference. In *Proceedings of the International Conference on Machine Learning*. PMLR, 3020–3029.

[15] Maria Kaselimi, Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Demitris Delikaraoglou. 2020. A causal long short-term memory sequence to sequence model for TEC prediction using GNSS observations. *Remote Sens.* 12, 9 (2020), 1354.

[16] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Retrieved from https://arXiv:1412.6980

[17] Kundan Krishna, Deepali Jain, Sanket V. Mehta, and Sunav Choudhary. 2018. An lstm based system for prediction of human activities with durations. *Proc. ACM Interact., Mobile, Wear. Ubiq. Technol.* 1, 4 (2018), 1–31.

[18] Young D. Kwon, Dimitris Chatzopoulos, Ehsan ul Haq, Raymond Chi-Wing Wong, and Pan Hui. 2019. GeoLifecycle: User engagement of geographical exploration and churn prediction in LBSNs. *Proc. ACM Interact., Mobile, Wear. Ubiq. Technol.* 3, 3 (2019), 1–29.

[19] Jia Li, Xiaowei Jia, Haoyu Yang, Vipin Kumar, Michael Steinbach, and Gyorgy Simon. 2020. Teaching deep learning causal effects improves predictive performance. Retrieved from https://arXiv:2011.05466

[20] Zongyu Lin, Shiqing Lyu, Hancheng Cao, Fengli Xu, Yuqiong Wei, Hanan Samet, and Yong Li. 2020. HealthWalks: Sensing fine-grained individual health condition via mobility data. *Proc. ACM Interact., Mobile, Wear. Ubiq. Technol.* 4, 4 (2020), 1–26.

[21] James McInerney, Brian Brost, Praveen Chandar, Rishabh Mehrotra, and Benjamin Carterette. 2020. Counterfactual evaluation of slate recommendations with sequential reward interactions. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1779–1788.

[22] Vishal Meshram, Vidula Meshram, and Kailas Patil. 2016. A survey on ubiquitous computing. *ICTACT J. Soft Comput.* 6, 2 (2016), 1130–1135.

[23] Austin Nichols. 2007. Causal inference with observational data. *Stata J.* 7, 4 (2007), 507–541.

[24] Lizhen Nie, Mao Ye, Qiang Liu, and Dan Nicolae. 2021. VCNet and functional targeted regularization for learning causal effects of continuous treatments. Retrieved from https://arXiv:2103.07861

[25] Judea Pearl. 2009. Causal inference in statistics: An overview. *Stat. Surveys* 3 (2009), 96–146.

[26] Sijie Ruan, Jie Bao, Yuxuan Liang, Ruiyuan Li, Tianfu He, Chuishi Meng, Yanhua Li, Yingcai Wu, and Yu Zheng. 2020. Dynamic Public Resource Allocation Based on Human Mobility Prediction. *Proc. ACM Interact., Mobile, Wear. Ubiq. Technol.* 4, 1 (2020), 1–22.

[27] Patrick Schwab, Lorenz Linhardt, Stefan Bauer, Joachim M. Buhmann, and Walter Karlen. 2020. Learning counterfactual representations for estimating individual dose-response curves. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5612–5619.

[28] Uri Shalit, Fredrik D. Johansson, and David Sontag. 2017. Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the International Conference on Machine Learning*. PMLR, 3076–3085.

[29] Rebecca Sharp, Mihai Surdeanu, Peter Jansen, Peter Clark, and Michael Hammond. 2016. Creating causal embeddings for question answering with minimal supervision. Retrieved from https://arXiv:1609.08097

[30] Zheyan Shen, Peng Cui, Tong Zhang, and Kun Kunag. 2020. Stable learning via sample reweighting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5692–5699.

[31] Qi Tang, Guoen Xia, Xianquan Zhang, and Feng Long. 2020. A customer churn prediction model based on XGBoost and MLP. In *Proceedings of the International Conference on Computer Engineering and Application (ICCEA'20).* IEEE, 608–612.

[32] Ian H. Witten, Eibe Frank, Mark A. Hall, Christopher J. Pal, and Mining Data. 2005. Practical machine learning tools and techniques. In *Data Mining*, Vol. 2. Elsevier Amsterdam, The Netherlands, 403–413.

[33] Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. 2021. A survey on causal inference. *ACM Trans. Knowl. Discov. Data.* 15, 5 (2021), 1–46.

[34] Jinsung Yoon, James Jordon, and Mihaela Van Der Schaar. 2018. GANITE: Estimation of individualized treatment effects using generative adversarial nets. In *Proceedings of the International Conference on Learning Representations.*

[35] Donghan Yu, Yong Li, Fengli Xu, Pengyu Zhang, and Vassilis Kostakos. 2018. Smartphone app usage prediction using points of interest. *Proc. ACM Interact., Mobile, Wear. Ubiq. Technol.* 1, 4 (2018), 1–21.

[36] Guozhen Zhang, Jinwei Zeng, Zhengyue Zhao, Depeng Jin, and Yong Li. 2022. A counterfactual modeling framework for churn prediction. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining.* 1424–1432.

[37] Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyan Shen. 2021. Deep stable learning for out-of-distribution generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 5372–5382.

[38] Yunke Zhang, Fengli Xu, Tong Li, Vassilis Kostakos, Pan Hui, and Yong Li. 2021. Passive health monitoring using large scale mobility data. *Proc. ACM Interact., Mobile, Wear. Ubiq. Technol.* 5, 1 (2021), 1–23.

[39] Ziyang Zhang, Lingye Tan, and Weiwei Jiang. 2022. Free-floating bike-sharing demand prediction with deep learning. *Int. J. Mach. Learn. Comput.* 12, 2 (2022).

[40] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2020. Disentangling user interest and popularity bias for recommendation with causal embedding. Retrieved from https://arXiv:2006.11011

[41] Hao Zou, Peng Cui, Bo Li, Zheyan Shen, Jianxin Ma, Hongxia Yang, and Yue He. 2020. Counterfactual prediction for bundle treatment. *Adv. Neural Info. Process. Syst.* 33 (2020).

Received 19 November 2022; revised 15 December 2023; accepted 3 February 2024