Learning Deep Features in Instrumental Variable Regression

Anonymous authors
Paper under double-blind review

Abstract

Instrumental variable (IV) regression is a standard strategy for learning causal relationships between confounded treatment and outcome variables by utilizing an instrumental variable, which is conditionally independent of the outcome given the treatment. In classical IV regression, learning proceeds in two stages: stage 1 performs linear regression from the instrument to the treatment; and stage 2 performs linear regression from the treatment to the outcome, conditioned on the instrument. We propose a novel method, *deep feature instrumental variable regression (DFIV)*, to address the case where relations between instruments, treatments, and outcomes may be nonlinear. In this case, deep neural nets are trained to define informative nonlinear features on the instruments and treatments. We propose an alternating training regime for these features to ensure good end-to-end performance when composing stages 1 and 2, thus obtaining highly flexible feature maps in a computationally efficient manner. DFIV outperforms recent state-of-the-art methods on challenging IV benchmarks, including settings involving high dimensional image data. DFIV also exhibits competitive performance in off-policy policy evaluation for reinforcement learning, which can be understood as an IV regression task.

1 Introduction

The aim of supervised learning is to obtain a model based on samples observed from some data generating process, and to then make predictions about new samples generated from the same distribution. If our goal is to predict the effect of our actions on the world, however, our aim becomes to assess the influence of interventions on this data generating process. To answer such causal questions, a supervised learning approach is inappropriate, since our interventions, called *treatments*, may affect the underlying distribution of the variable of interest, which is called the *outcome*.

To answer these counterfactual questions, we need to learn how treatment variables causally affect the distribution process of outcomes, which is expressed in a *structural function*. Learning a structural function from observational data (that is, data where we can observe, but not intervene) is known to be challenging if there exists an unmeasured confounder, which influences both treatment and outcome. To illustrate: suppose we are interested in predicting sales of airplane tickets given price. During the holiday season, we would observe the simultaneous increase in sales and prices. This does not mean that raising the prices *causes* the sales to increase. In this context, the time of the year is a confounder, since it affects both the sales and the prices, and we need to correct the bias caused by it.

One way of correcting such bias is via *instrumental variable* (IV) regression (Stock and Trebbi, 2003). Here, the structural function is learned using instrumental variables, which only affect the treatment directly but not the outcome. In the sales prediction scenario, we can use supply cost shifters as the instrumental variable since they only affect the price (Wright, 1928; Blundell et al., 2012). Instrumental variables can be found in many contexts, and IV regression is extensively used by economists and epidemiologists. For example, IV regression is used for measuring the effect of a drug in the scenario of imperfect compliance (Angrist et al., 1996), or the influence of military service on lifetime earnings (Angrist, 1990).

In this work, we propose a novel IV regression method, which can discover non-linear causal relationships using deep neural networks.

Classically, IV regression is solved by the *two-stage least squares* (2SLS) algorithm; we learn a mapping from the instrument to the treatment in the first stage and learn the structural function in the second stage as the mapping from the conditional expectation of the treatment given the instrument (obtained from stage 1) to the outcome. Originally, 2SLS assumes linear relationships in both stages, but this has been recently extended to non-linear settings.

One approach has been to use non-linear feature maps. Sieve IV (Newey and Powell, 2003; Chen and Christensen, 2018) uses a finite number of basis functions explicitly specified. Kernel IV (KIV) (Singh et al., 2019) and Dual IV regression (Muandet et al., 2019) extend sieve IV to allow for an infinite number of basis functions using reproducing kernel Hibert spaces (RKHS). Although these methods enjoy desirable theoretical properties, the flexibility of the model is limited, since all existing work uses the prespecified features.

Another approach is to perform the stage 1 regression through conditional density estimation (Carrasco et al., 2007; Darolles et al., 2011; Hartford et al., 2017). One advantage of this approach is that it allows for flexible models, including deep neural nets, as proposed in the DeepIV algorithm of (Hartford et al., 2017). It is known, however, that conditional density estimation is costly and often suffers from high variance when the treatment is high-dimensional. This issue is known as "forbidden regression", and is discussed in Angrist and Pischke (2009, §4.6.1) and Bennett et al. (2019).

To mitigate the "forbidden regression" problem, Bennett et al. (2019) propose DeepGMM, a method inspired by the optimally weighted Generalized Method of Moments (GMM) (Hansen, 1982) to find a structural function ensuring that the regression residual and the instrument are independent. Although this approach can handle high-dimensional treatment variables and deep NNs as feature extractors, the learning procedure might not be as stable as 2SLS approach, since it involves solving a smooth zero-sum game.

In this paper, we propose *Deep Feature Instrumental Variable Regression* (DFIV), which aims to combine the advantages of all previous approaches, while avoiding their limitations. In DFIV, we use deep neural nets to adaptively learn feature maps in the 2SLS approach, which allows us to fit highly nonlinear structural functions, as in DeepGMM and DeepIV. DFIV avoids the "forbidden regression" issue in DeepIV, however, since it does not rely on conditional density estimation. As in sieve IV and KIV, DFIV learns the conditional expectation of the feature maps in stage 1 and uses the predicted features in stage 2, but with the additional advantage of learned features. We empirically show that DFIV performs better than other methods on several IV benchmarks, and apply DFIV successfully to off-policy policy evaluation, which is a fundamental problem in Reinforcement Learning (RL).

The paper is structured as follows. In Section 2, we formulate the IV regression problem and introduce two-stage least-squares regression. In Section 3, we give a detailed description of our DFIV method. We demonstrate the empirical performance of DFIV in Section 4, covering three settings: a classical demand prediction example from econometrics, a challenging IV setting where the treatment consists of high-dimensional image data, and the problem of off-policy policy evaluation in reinforcement learning.

2 Preliminaries

2.1 Problem Setting of Instrumental Variable Regression

We begin with a description of the IV setting. We observe a treatment $X \in \mathcal{X}$, where $\mathcal{X} \subset \mathbb{R}^{d_X}$, and an outcome $Y \in \mathcal{Y}$, where $\mathcal{Y} \subset \mathbb{R}$. an unobserved confounder that affects both X and Y. This causal relationship can be represented with the following structural causal model:

$$Y = f_{\text{struct}}(X) + \varepsilon, \quad \mathbb{E}[\varepsilon] = 0, \quad \mathbb{E}[\varepsilon | X] \neq 0, \tag{1}$$

where f_{struct} is called the structural function, which we assume to be a continuous, and ε is an additive noise term. The

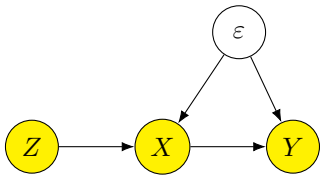

Figure 1: Causal Graph.

corresponding graphical model is shown in Figure 1. The challenge is that $\mathbb{E}[\varepsilon|X] \neq 0$, which reflects the existence of a confounder. Because of this, we cannot use ordinary supervised learning techniques since $f_{\text{struct}}(x) \neq \mathbb{E}[Y|X=x]$. Here, we assume there is no observable confounder but we may easily include this, as discussed in Appendix B.

To deal with the hidden confounder ε, we assume to have access to an instrumental variable $Z \in \mathcal{Z}$ which satisfies the following assumption.

Assumption 1. *The conditional distribution $P(X|Z)$ is not constant in Z and $\mathbb{E}[\varepsilon|Z] = 0$.*

Intuitively, Assumption 1 means that the instrument Z induces variation in the treatment X but is uncorrelated with the hidden confounder ε. Again, for simplicity, we assume $\mathcal{Z} \subset \mathbb{R}^{d_Z}$. Given Assumption 1, we can see that the function f_{struct} satisfies the operator equation $\mathbb{E}[Y|Z] = \mathbb{E}[f_{\text{struct}}(X)|Z]$ by taking expectation conditional on Z of both sides of (1). Solving this equation, however, is known to be ill-posed (Nashed and Wahba, 1974). To address this, recent works (Carrasco et al., 2007; Darolles et al., 2011; Muandet et al., 2019; Singh et al., 2019) minimize the following regularized loss $\mathcal{L}$ to obtain the estimate $\hat{f}_{\text{struct}}$:

$$\hat{f}_{\text{struct}} = \underset{f \in \mathcal{F}}{\arg\min}\, \mathcal{L}(f), \quad \mathcal{L}(f) = \mathbb{E}_{YZ}\left[(Y - \mathbb{E}_{X|Z}[f(X)])^2\right] + \Omega(f), \tag{2}$$

where $\mathcal{F}$ is an arbitrary space of continuous functions and $\Omega(f)$ is a regularizer on f.

2.2 Two Stage Least Squares Regression

A number of works (Newey and Powell, 2003; Singh et al., 2019) tackle the minimization problem (2) using two-stage least squares (2SLS) regression, in which the structural function is modeled as $f_{\text{struct}}(x) = \boldsymbol{u}^\top \boldsymbol{\psi}(x)$, where $\boldsymbol{u}$ is a learnable weight vector and $\boldsymbol{\psi}(x)$ is a vector of fixed basis functions. For example, linear 2SLS used the identity map $\boldsymbol{\psi}(x) = x$, while sieve IV (Newey and Powell, 2003) uses Hermite polynomials.

In the 2SLS approach, an estimate $\hat{\boldsymbol{u}}$ is obtained by solving two regression problems successively. In stage 1, we estimate the conditional expectation $\mathbb{E}_{X|z}[\boldsymbol{\psi}(X)]$ as a function of z. Then in stage 2, as $\mathbb{E}_{X|z}[f(X)] = \boldsymbol{u}^\top \mathbb{E}_{X|z}[\boldsymbol{\psi}(X)]$, we minimize $\mathcal{L}$ with $\mathbb{E}_{X|z}[f(X)]$ being replaced by the estimate obtained in stage 1.

Specifically, we model the conditional expectation as $\mathbb{E}_{X|z}[\boldsymbol{\psi}(X)] = \boldsymbol{V}\boldsymbol{\phi}(z)$, where $\boldsymbol{\phi}(z)$ is another vector of basis functions and $\boldsymbol{V}$ is a *matrix* to be learned. Again, there exist many choices for $\boldsymbol{\phi}(z)$, which can be infinite-dimensional, but we assume the dimensions of $\boldsymbol{\psi}(x)$ and $\boldsymbol{\phi}(z)$ to be $d_1, d_2 < \infty$ respectively.

In stage 1, the matrix $\boldsymbol{V}$ is learned by minimizing the following loss,

$$\hat{\boldsymbol{V}} = \underset{\boldsymbol{V} \in \mathbb{R}^{d_1 \times d_2}}{\arg\min}\, \mathcal{L}_1(\boldsymbol{V}), \quad \mathcal{L}_1(\boldsymbol{V}) = \mathbb{E}_{X,Z}\left[\|\boldsymbol{\psi}(X) - \boldsymbol{V}\boldsymbol{\phi}(Z)\|^2\right] + \lambda_1 \|\boldsymbol{V}\|^2, \tag{3}$$

where $\lambda_1 > 0$ is a regularization parameter. This is a linear ridge regression problem with multiple targets, which can be solved analytically. In stage 2, given $\hat{\boldsymbol{V}}$, we can obtain $\boldsymbol{u}$ by minimizing the loss

$$\hat{\boldsymbol{u}} = \underset{\boldsymbol{u} \in \mathbb{R}^{d_1}}{\arg\min}\, \mathcal{L}_2(\boldsymbol{u}), \quad \mathcal{L}_2(\boldsymbol{u}) = \mathbb{E}_{Y,Z}\left[\|Y - \boldsymbol{u}^\top \hat{\boldsymbol{V}}\boldsymbol{\phi}(Z)\|^2\right] + \lambda_2 \|\boldsymbol{u}\|^2, \tag{4}$$

where $\lambda_2 > 0$ is another regularization parameter. Stage 2 corresponds to a ridge linear regression from $\hat{\boldsymbol{V}}\boldsymbol{\phi}(Z)$ to Y, and also enjoys a closed-form solution. Given the learned weights $\hat{\boldsymbol{u}}$, the estimated structural function is $\hat{f}_{\text{struct}}(x) = \hat{\boldsymbol{u}}^\top \boldsymbol{\psi}(x)$.

3 DFIV Algorithm

In this section, we develop the DFIV algorithm. Similarly to Singh et al. (2019), we assume that we do not necessarily have access to observations from the joint distribution of (X, Y, Z). Instead, we are given m observations of (X, Z) for stage 1 and n observations of (Y, Z) for stage 2. We denote the stage 1 observations as (x_i, z_i) and the stage 2 observations as $(\tilde{y}_i, \tilde{z}_i)$.

If observations of (X, Y, Z) are given for both stages, we can evaluate the out-of-sample losses, and these losses can be used for hyper-parameter tuning of λ_1, λ_2 (Appendix A).

DFIV uses the following models

$$f_{\text{struct}}(x) = \boldsymbol{u}^\top \boldsymbol{\psi}_{\theta_X}(x) \quad \text{and} \quad \mathbb{E}_{X|z}\left[\boldsymbol{\psi}_{\theta_X}(X)\right] = \boldsymbol{V}\boldsymbol{\phi}_{\theta_Z}(z), \tag{5}$$

where $\boldsymbol{u} \in \mathbb{R}^{d_1}$ and $\boldsymbol{V} \in \mathbb{R}^{d_1 \times d_2}$ are the parameters, and $\boldsymbol{\psi}_{\theta_X}(x) \in \mathbb{R}^{d_1}$ and $\boldsymbol{\phi}_{\theta_Z}(z) \in \mathbb{R}^{d_2}$ are the neural nets parameterised by θ_X and θ_Z, respectively. As in the original 2SLS algorithm, we learn $\mathbb{E}_{X|z}\left[\boldsymbol{\psi}_{\theta_X}(X)\right]$ in stage 1 and $f_{\text{struct}}(x)$ in stage 2. In addition to the weights $\boldsymbol{u}$ and $\boldsymbol{V}$, however, we also learn the parameters of the feature maps, θ_X and θ_Z. Hence, we need to alternate between stages 1 and 2, since the conditional expectation $\mathbb{E}_{X|z}\left[\boldsymbol{\psi}_{\theta_X}(X)\right]$ changes during training.

Stage 1 Regression The goal of stage 1 is to estimate the conditional expectation $\mathbb{E}_{X|z}\left[\boldsymbol{\psi}_{\theta_X}(X)\right] \simeq \boldsymbol{V}\boldsymbol{\psi}_{\theta_Z}(z)$ by learning the matrix $\boldsymbol{V}$ and parameter θ_Z, with $\theta_X = \hat{\theta}_X$ given and fixed. Given the stage 1 data (x_i, z_i), this can be done by minimizing the empirical estimate of $\mathcal{L}_1$,

$$\hat{\boldsymbol{V}}^{(m)}, \hat{\theta}_Z = \underset{\boldsymbol{V}, \theta_Z}{\arg\min}\, \mathcal{L}_1^{(m)}(\boldsymbol{V}, \theta_Z), \;\; \mathcal{L}_1^{(m)} = \frac{1}{m}\sum_{i=1}^{m} \|\boldsymbol{\psi}_{\hat{\theta}_X}(x_i) - \boldsymbol{V}\boldsymbol{\phi}_{\theta_Z}(z_i)\|^2 + \lambda_1 \|\boldsymbol{V}\|^2. \tag{6}$$

Note that the feature map $\boldsymbol{\psi}_{\hat{\theta}_X}(X)$ is fixed during stage 1, since this is the "target variable." If we fix θ_Z, the minimization problem (6) reduces to a linear ridge regression problem with multiple targets, whose solution as a function of θ_X and θ_Z is given analytically by

$$\hat{\boldsymbol{V}}^{(m)}(\theta_X, \theta_Z) = \Psi_1^\top \Phi_1 (\Phi_1^\top \Phi_1 + m\lambda_1 I)^{-1}, \tag{7}$$

where Φ_1, Ψ_1 are feature matrices defined as $\Psi_1 = [\boldsymbol{\psi}_{\theta_X}(x_1), \ldots, \boldsymbol{\psi}_{\theta_X}(x_m)]^\top \in \mathbb{R}^{m \times d_1}$ and $\Phi_1 = [\boldsymbol{\phi}_{\theta_Z}(z_1), \ldots, \boldsymbol{\phi}_{\theta_Z}(z_m)]^\top \in \mathbb{R}^{m \times d_2}$. We can then learn the parameters θ_Z of the adaptive features $\boldsymbol{\psi}_{\theta_Z}$ by minimizing the loss $\mathcal{L}_1^{(m)}$ at $\boldsymbol{V} = \hat{\boldsymbol{V}}^{(m)}(\hat{\theta}_X, \theta_Z)$ using gradient descent. For simplicity, we introduce a small abuse of notation by denoting as $\hat{\theta}_Z$ the result of a user-chosen number of gradient descent steps on the loss (6) with $\hat{\boldsymbol{V}}^{(m)}(\hat{\theta}_X, \theta_Z)$ from (7), even though $\hat{\theta}_Z$ need not attain the minimum of the non-convex loss (6). We then write $\hat{\boldsymbol{V}}^{(m)} := \hat{\boldsymbol{V}}^{(m)}(\hat{\theta}_X, \hat{\theta}_Z)$. While this trick of using an analytical estimate of the linear output weights of a deep neural network might not lead to significant gains in standard supervised learning, it turns out to be very important in the development of our 2SLS algorithm. As shown in the following section, the analytical estimate $\hat{\boldsymbol{V}}^{(m)}(\theta_X, \hat{\theta}_Z)$ (now considered as a function of θ_X) will be used to backpropagate to θ_X in stage 2.

Stage 2 Regression In stage 2, we learn the structural function by computing the weight vector $\boldsymbol{u}$ and parameter θ_X while *fixing* $\theta_Z = \hat{\theta}_Z$, and thus the corresponding feature map $\boldsymbol{\phi}_{\hat{\theta}_Z}(z)$. Given the data $(\tilde{y}_i, \tilde{z}_i)$, we can minimize the empirical version of $\mathcal{L}_2$, defined as

$$\hat{\boldsymbol{u}}^{(n)}, \hat{\theta}_X = \underset{\boldsymbol{u} \in \mathbb{R}^{d_1}, \theta_X}{\arg\min}\, \mathcal{L}_2^{(n)}(\boldsymbol{u}, \theta_X), \quad \mathcal{L}_2^{(n)} = \frac{1}{n}\sum_{i=1}^{n} (\tilde{y}_i - \boldsymbol{u}^\top \hat{\boldsymbol{V}}^{(m)} \boldsymbol{\phi}_{\hat{\theta}_Z}(\tilde{z}_i))^2 + \lambda_2 \|\boldsymbol{u}\|^2. \tag{8}$$

Again, for a given θ_X, we can solve the minimization problem (8) for $\boldsymbol{u}$ as a function of $\hat{\boldsymbol{V}}^{(m)} := \hat{\boldsymbol{V}}^{(m)}(\theta_X, \hat{\theta}_Z)$ by a linear ridge regression

$$\hat{\boldsymbol{u}}^{(n)}(\theta_X, \hat{\theta}_Z) = \left(\hat{\boldsymbol{V}}^{(m)} \Phi_2^\top \Phi_2 (\hat{\boldsymbol{V}}^{(m)})^\top + n\lambda_2 I\right)^{-1} \hat{\boldsymbol{V}}^{(m)} \Phi_2^\top \boldsymbol{y}_2, \tag{9}$$

where $\Phi_2 = [\boldsymbol{\phi}_{\hat{\theta}_Z}(\tilde{z}_1), \ldots, \boldsymbol{\phi}_{\hat{\theta}_Z}(\tilde{z}_n)]^\top \in \mathbb{R}^{n \times d_2}$ and $\boldsymbol{y}_2 = [\tilde{y}_1, \ldots, \tilde{y}_n]^\top \in \mathbb{R}^n$.

The loss $\mathcal{L}_2^{(n)}$ explicitly depends on the parameters θ_X and we can backpropagate it to θ_X via $\hat{\boldsymbol{V}}^{(m)}(\theta_X, \hat{\theta}_Z)$, even though the samples of the treatment variable X do not appear in stage 2 regression. We again introduce a small abuse of notation for simplicity, and denote by $\hat{\theta}_X$ the estimate obtained after a few gradient steps on (8) with $\hat{\boldsymbol{u}}^{(n)}(\theta_X, \hat{\theta}_Z)$ from (9), even

though $\hat{\theta}_X$ need not minimize the non-convex loss (8). We then have $\hat{\boldsymbol{u}}^{(n)} = \hat{\boldsymbol{u}}^{(n)}(\hat{\theta}_X, \hat{\theta}_Z)$. After updating $\hat{\theta}_X$, we need to update $\hat{\theta}_Z$ accordingly. We do not attempt to backpropagate through the estimate $\hat{\theta}_Z$ to do this, however, as this would be too computationally expensive; instead, we alternate stages 1 and 2. We also considered updating $\hat{\theta}_X$ and $\hat{\theta}_Z$ jointly to optimize the loss $\mathcal{L}_2^{(n)}$, but this fails, as discussed in Appendix E.

Computational Complexity The computational complexity of the algorithm is $O(md_1d_2+d_2^3)$ for stage 1, while stage 2 requires additional $O(nd_1d_2+d_1^3)$ computations. This is small compared to KIV (Singh et al., 2019), which takes $O(m^3)$ and $O(n^3)$, respectively. We can further speed up the learning by using mini-batch training as shown in Algorithm 1.

Algorithm 1 Deep Feature Instrumental Variable Regression

Input: Stage 1 data (x_i, z_i), Stage 2 data $(\tilde{y}_i\tilde{z}_i)$, Regularization parameters (λ_1, λ_2). Initial values $\hat{\theta}_X, \hat{\theta}_Z$. Mini-batch size (m_b, n_b). Number of updates in each stage (T_1, T_2).

Output: Estimated structural function $\hat{f}_{\text{struct}}(x)$

repeat
  Sample m_b stage 1 data $(x_i^{(b)}, z_i^{(b)})$ and n_b stage 2 data $(\tilde{y}_i^{(b)}, \tilde{z}_i^{(b)})$.
  for $t = 1$ to T_1 **do**
    Return function $\hat{\boldsymbol{V}}^{(m_b)}(\hat{\theta}_X, \theta_Z)$ in (7) using $(x_i^{(b)}, z_i^{(b)})$
    Update $\hat{\theta}_Z \leftarrow \hat{\theta}_Z - \alpha\nabla_{\theta_Z}\mathcal{L}_1^{(m_b)}(\hat{\boldsymbol{V}}^{(m_b)}(\hat{\theta}_X, \theta_Z), \theta_Z)|_{\theta_Z=\hat{\theta}_Z}$ \\ *Stage 1 learning*
  end for
  for $t = 1$ to T_2 **do**
    Return function $\hat{\boldsymbol{u}}^{(n_b)}(\theta_X, \hat{\theta}_Z)$ in (9) using $(\tilde{y}_i^{(b)}, \tilde{z}_i^{(b)})$ and function $\hat{\boldsymbol{V}}^{(m_b)}(\theta_X, \hat{\theta}_Z)$
    Update $\hat{\theta}_X \leftarrow \hat{\theta}_X - \alpha\nabla_{\theta_X}\mathcal{L}_2^{(n_b)}(\hat{\boldsymbol{u}}^{(n_b)}(\theta_X, \hat{\theta}_Z), \theta_X)|_{\theta_X=\hat{\theta}_X}$ \\ *Stage 2 learning*
  end for
until convergence
Compute $\hat{\boldsymbol{u}}^{(n)} := \hat{\boldsymbol{u}}^{(n)}(\hat{\theta}_X, \hat{\theta}_Z)$ from (9) using entire dataset.
return $\hat{f}_{\text{struct}}(x) = (\hat{\boldsymbol{u}}^{(n)})^\top \boldsymbol{\psi}_{\hat{\theta}_X}(x)$

Here, $\hat{\boldsymbol{V}}^{(m_b)}$ and $\hat{\boldsymbol{u}}^{(n_b)}$ are the functions given by (7) and (9) calculated using mini-batches of data. Similarly, $\mathcal{L}_1^{(m_b)}$ and $\mathcal{L}_1^{(n_b)}$ are the stage 1 and 2 losses for the mini-batches. We recommend setting the batch size large enough so that $\hat{\boldsymbol{V}}^{(m_b)}, \hat{\boldsymbol{u}}^{(n_b)}$ do not diverge from $\hat{\boldsymbol{V}}^{(m)}, \hat{\boldsymbol{u}}^{(n)}$ computed on the entire dataset. Furthermore, we observe that setting $T_1 > T_2$, i.e. updating θ_Z more frequently than θ_X, stabilizes the learning process.

4 Experiments

In this section, we report the empirical performance of the DFIV method. The evaluation considers both low and high-dimensional treatment variables. We used the demand design dataset of Hartford et al. (2017) for benchmarking in the low and high-dimensional cases, and we propose a new setting for the high-dimensional case based on the dSprites dataset (Matthey et al., 2017). In the deep RL context, we also apply DFIV to perform off-policy policy evaluation (OPE). The network architecture and hyper-parameters are provided in Appendix F. The algorithms in the first two experiments are implemented using PyTorch (Paszke et al., 2019) and the OPE experiments are implemented using TensorFlow (Abadi et al., 2015) and the Acme RL framework (Hoffman et al., 2020). The code is included in the supplemental material.

4.1 Demand Design Experiments

The demand design dataset is a synthetic dataset introduced by Hartford et al. (2017) that is now a standard benchmarking dataset for testing nonlinear IV methods. In this dataset, we aim to predict the demands on airplane tickets Y given the price of the tickets P. The dataset contains two observable confounders, which are the time of year $T \in [0, 10]$ and

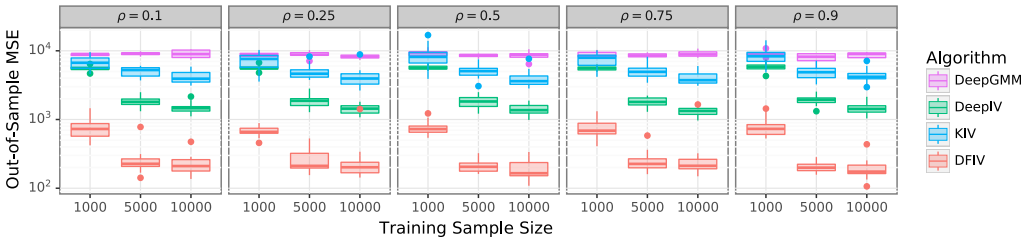

Figure 2: MSE for demand design dataset with low dimensional confounders.

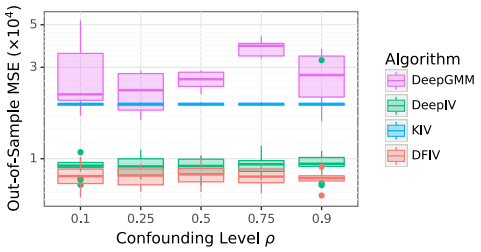

Figure 3: MSE for demand design dataset with high dimensional observed confounders.

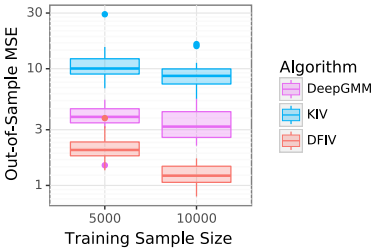

Figure 4: MSE for dSprite dataset. DeepIV did not yield meaningful predictions for this experiment.

customer groups $S \in \{1, ..., 7\}$ that are categorized by the levels of price sensitivity. Further, the noise in Y and P is correlated, which indicates the existence of an unobserved confounder. The strength of the correlation is represented by $\rho \in [0, 1]$. To correct the bias caused by this hidden confounder, the fuel price C is introduced as an instrumental variable. Details of the data generation process can be found in Appendix D.1. In DFIV notation, the treatment is $X = P$, the instrument is $Z = C$, and (T, S) are the observable confounders.

We compare the DFIV method to three leading modern competitors, namely KIV (Singh et al., 2019), DeepIV (Hartford et al., 2017), and DeepGMM (Bennett et al., 2019). We used the DFIV method with observable confounders, as introduced in Appendix B. Note that DeepGMM does not have an explicit mechanism for incorporating observable confounders. The solution we use, proposed by Bennett et al. (2019, p. 2), is to incorporate these observables in both instrument *and* treatment; hence we apply DeepGMM with treatment $X = (P, T, S)$ and instrumental variable $Z = (C, T, S)$. Although this approach is theoretically sound, this makes the problem unnecessary difficult since it ignores the fact that we only need to consider the conditional expectation of P given Z.

We used a network with a similar number of parameters to DeepIV as the feature maps in DFIV and models in DeepGMM. We tuned the regularizers λ_1, λ_2 as discussed in Appendix A, with the data evenly split for stage 1 and stage 2. We varied the correlation parameter ρ and dataset size, and ran 20 simulations for each setting. Results are summarized in Figure 2.

Next, we consider a case, introduced by Hartford et al. (2017), where the customer type $S \in \{1, \ldots, 7\}$ is replaced with an image of the corresponding handwritten digit from the MNIST dataset (LeCun and Cortes, 2010). This reflects the fact that we cannot know the exact customer type, and thus we need to estimate it from noisy high-dimensional data. Note that although the confounder is high-dimensional, the treatment variable is still real-valued, i.e. the price P of the tickets. Figure 3 presents the results for this high-dimensional confounding case. Again, we train the networks with a similar number of learnable parameters to DeepIV in DFIV and DeepGMM, and hyper-parameters are set in the way discussed in Appendix A. We ran 20 simulations with data size $n + m = 5000$ and report the mean and standard error.

Our first observation from Figure 2 and 3 is that the level ρ of correlation has no significant impact on the error under any of the IV methods, indicating that all approaches correctly account for the effect of the hidden confounder. This is consistent with earlier results on

this dataset using KIV and DeepIV (Singh et al., 2019; Hartford et al., 2017). We note that DeepGMM does not perform well in this demand design problem. This may be due to the current DeepGMM approach to handling observable confounders, which might not be optimal. KIV performed reasonably well for small sample sizes and low-dimensional data, but it did less well in the high-dimensional MNIST case due to its less expressive features. In high dimensions, DeepIV performed well, since the treatment variable was unidimensional and the "forbidden regression" issue did not arise. DFIV performed consistently better than all other methods in both low and high dimensions, which suggests it can stably learn a flexible structural function.

4.2 dSprites Experiments

To test the performance of DFIV methods for a high dimensional treatment variable, we utilized the dSprites dataset (Matthey et al., 2017). This is an image dataset described by five latent parameters (`shape, scale, rotation, posX` and `posY`). The images are $64 \times 64 = 4096$-dimensional. In this experiment, we fixed the `shape` parameter to `heart`, i.e. we only used heart-shaped images. An example is shown in Figure 5.

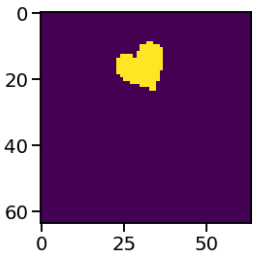

Figure 5: dSprite image

From this dataset, we generated data for IV regression in which we use each figure as treatment variable X. Hence, the treatment variable is 4096-dimensional in this experiment. To make the task more challenging, we used `posY` as the hidden confounder, which is not revealed to the model. We used the other three latent variables as the instrument variables Z. See Appendix D.2 for the detailed data generation process.

We tested the performance of DFIV with KIV and DeepGMM, where the hyper-parameters are determined as in the demand design problem. The results are displayed in Figure 4. DFIV consistently yields the best performance of all the methods. DeepIV is not included in the figure because it fails to give meaningful predictions due to the "forbidden regression" issue, as the treatment variable is high-dimensional. The performance of KIV suffers since it lacks the feature richness to express a high-dimensional complex structural function.

4.3 Off-Policy Policy Evaluation Experiments

We apply our IV methods to the off-policy policy evaluation (OPE) problem (Sutton and Barto, 2018), which is one of the fundamental problems of deep RL. In particular, it has been realized by Bradtke and Barto (1996) that 2SLS could be used to estimate a linearly parameterized value function, and we use this reasoning as the basis of our approach. Let us consider the RL environment $\langle \mathcal{S}, \mathcal{A}, \rho_0, P, R, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\rho_0 : \mathcal{S} \to [0, 1]$ is the initial state distribution, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function, $R : \mathcal{S} \times \mathcal{A} \times \mathbb{R} \to \mathbb{R}$ is the reward distribution, and discount factor $\gamma \in (0, 1]$. Let π be a policy, and we denote $\pi(a|s)$ as the probability of selecting action a in stage $s \in \mathcal{S}$. Given policy π, the Q-function is defined as

$$Q^\pi(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \middle| s_0 = s, a_0 = a\right]$$

with $a_t \sim \pi(\cdot \mid s_t), s_{t+1} \sim P(\cdot|s_t, a_t), r_t \sim R(\cdot|s_t, a_t)$. The goal of OPE is to evaluate the expectation of the Q-function with respect to the initial state distribution for a given target policy π, $\mathbb{E}_{s\sim\rho_0, a|s\sim\pi}[Q^\pi(s, a)]$, learned from a fixed dataset of transitions (s, a, r, s'), where s and a are sampled from some potentially unknown distribution μ and behavioral policy $\pi_b(\cdot|s)$ respectively. Using the Bellman equation satisfied by Q^π, we obtain a structural causal model of the form (1),

$$\begin{aligned} r = \quad & \overbrace{Q^\pi(s, a) - \gamma Q^\pi(s', a')}^{\text{structural function } f_{\text{struct}}(s,a,s',a')} \\ & + \underbrace{\gamma \left(Q^\pi(s', a') - \mathbb{E}_{s'\sim P(\cdot|s,a), a'\sim\pi(\cdot|s')} \left[Q^\pi(s', a')\right]\right) + r - \mathbb{E}_{r\sim R(\cdot|s,a)}\left[r\right]}_{\text{confounder } \varepsilon}, \end{aligned} \tag{10}$$

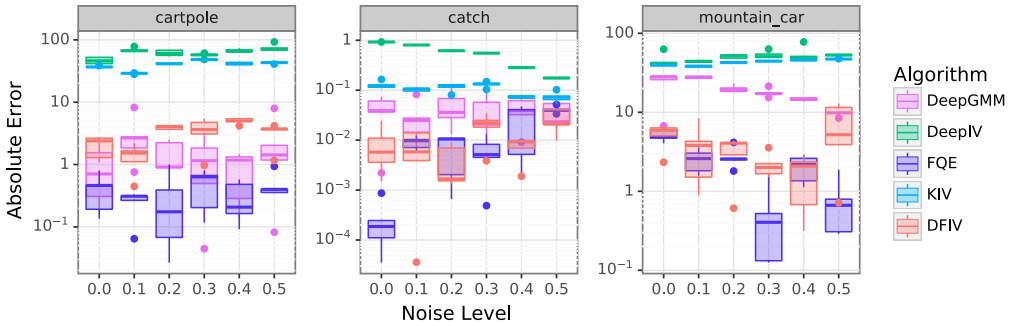

Figure 6: Error of off-policy policy evaluation.

where $X = (s, a, s', a'), Z = (s, a), Y = r$. We have that $\mathbb{E}[\varepsilon] = 0, \mathbb{E}[\varepsilon|X] \neq 0$, and Assumption 1 is verified. Minimizing the loss (2) for the structural causal model (10) corresponds to minimizing the mean squared Bellman error (MSBE), defined as

$$\text{MSBE} = \mathbb{E}_{s,a}\left[\left(r + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a), a' \sim \pi(\cdot|s')}\left[Q^\pi(s', a')\right] - Q^\pi(s, a)\right)^2\right], \tag{11}$$

and we can apply any IV regression method to achieve this. Note that MBSE is also the loss minimized by the residual gradient (RG) method proposed in (Baird, 1995) to estimate Q-functions. However, this method suffers from the "double-sample" issue, i.e. it requires two independent samples of s' starting from the same (s, a) due to the inner conditional expectation (Baird, 1995), whereas IV regression methods do not suffer from this issue.

The detailed DFIV algorithm applied to OPE is given in Appendix C. We evaluate DFIV on three BSuite (Osband et al., 2019) tasks: catch, mountain car, and cartpole. The original system dynamics are deterministic. To create a stochastic environment, we randomly replace the agent action by a uniformly sampled action with probability $p \in [0, 0.5]$. The noise level p controls the level of confounding effect. The target policy is trained using DQN (Mnih et al., 2015), and we subsequently generate an offline dataset for OPE by executing the policy in the same environment with a random action probability of 0.2 (on top of the environment's random action probability p). We compare DFIV with KIV, DeepIV, and DeepGMM; as well as Fitted Q Evaluation (FQE) (Le et al., 2019; Voloshin et al., 2019), a specialized approach designed for the OPE setting, which serves as our "gold standard" baseline (Paine et al., 2020). All methods use the same network for value estimation. Figure 6 shows the absolute error of the estimated policy value by each method with a standard deviation from 5 runs. In catch and mountain car, DFIV comes closest in performance to FQE, and even matches it for some noise settings, whereas DeepGMM is somewhat worse in catch, and significantly worse in mountain car. In the case of cartpole, DeepGMM performs somewhat better than DFIV, although both are slightly worse than FQE. DeepIV and KIV both do poorly across all RL benchmarks.

5 Conclusion

We have proposed a novel method for instrumental variable regression, Deep Feature IV (DFIV), which performs two-stage least squares regression on flexible and expressive features of the instrument and treatment. As a contribution to the IV literature, we showed how to adaptively learn these feature maps with deep neural networks. We also showed that the off-policy policy evaluation (OPE) problem in deep RL can be interpreted as a nonlinear IV regression, and that DFIV performs competitively in this domain. In RL problems with additional external confounders, we believe DFIV will be of great value. This work brings the research worlds of deep offline RL and causality from observational data closer.

References

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia,

R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL http://tensorflow.org/.

J. D. Angrist. Lifetime earnings and the Vietnam era draft lottery: Evidence from social security administrative records. *The American Economic Review*, 80(3):313–336, 1990.

J. D. Angrist and J.-S. Pischke. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton University Press, 1st edition, 2009.

J. D. Angrist, G. W. Imbens, and D. B. Rubin. Identification of causal effects using instrumental variables. *Journal of the American Statistical Association*, 91(434):444–455, 1996.

L. Baird. Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Machine Learning*, 1995.

A. Bennett, N. Kallus, and T. Schnabel. Deep generalized method of moments for instrumental variable analysis. In *Advances in Neural Information Processing Systems 32*, pages 3564–3574. 2019.

R. Blundell, J. Horowitz, and M. Parey. Measuring the price responsiveness of gasoline demand: Economic shape restrictions and nonparametric demand estimation. *Quantitative Economics*, 3:29–51, 2012.

S. J. Bradtke and A. G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(1-3):33–57, 1996.

M. Carrasco, J.-P. Florens, and E. Renault. Linear inverse problems in structural econometrics estimation based on spectral decomposition and regularization. In *Handbook of Econometrics*, volume 6B, chapter 77. 2007.

X. Chen and T. M. Christensen. Optimal sup-norm rates and uniform inference on nonlinear functionals of nonparametric IV regression: Nonlinear functionals of nonparametric IV. *Quantitative Economics*, 9:39–84, 2018.

S. Darolles, Y. Fan, J. P. Florens, and E. Renault. Nonparametric instrumental regression. *Econometrica*, 79(5):1541–1565, 2011.

L. P. Hansen. Large sample properties of generalized method of moments estimators. *Econometrica*, 50(4):1029–1054, 1982.

J. Hartford, G. Lewis, K. Leyton-Brown, and M. Taddy. Deep IV: A flexible approach for counterfactual prediction. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

M. Hoffman, B. Shahriari, J. Aslanides, G. Barth-Maron, F. Behbahani, T. Norman, A. Abdolmaleki, A. Cassirer, F. Yang, K. Baumli, S. Henderson, A. Novikov, S. G. Colmenarejo, S. Cabi, C. Gulcehre, T. L. Paine, A. Cowie, Z. Wang, B. Piot, and N. de Freitas. Acme: A research framework for distributed reinforcement learning. *arXiv preprint arXiv:2006.00979*, 2020. URL https://arxiv.org/abs/2006.00979.

H. M. Le, C. Voloshin, and Y. Yue. Batch policy learning under constraints. *arXiv preprint arXiv:1903.08738*, 2019.

Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner. dSprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

K. Muandet, A. Mehrjou, S. K. Lee, and A. Raj. Dual IV: A single stage instrumental variable regression. *arXiv preprint arXiv:1910.12358*, 2019.

M. Z. Nashed and G. Wahba. Generalized inverses in reproducing kernel spaces: An approach to regularization of linear operator equations. *SIAM Journal on Mathematical Analysis*, 5 (6):974–987, 1974.

W. K. Newey and J. L. Powell. Instrumental variable estimation of nonparametric models. *Econometrica*, 71(5):1565–1578, 2003.

I. Osband, Y. Doron, M. Hessel, J. Aslanides, E. Sezener, A. Saraiva, K. McKinney, T. Lattimore, C. Szepesvari, S. Singh, et al. Behaviour suite for reinforcement learning. In *International Conference on Learning Representations*, 2019.

T. L. Paine, C. Paduraru, A. Michi, C. Gulcehre, K. Zolna, A. Novikov, Z. Wang, and N. de Freitas. Hyperparameter selection for offline reinforcement learning. *arXiv preprint arXiv:2007.09055*, 2020.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. 2019.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20*, pages 1177–1184. 2008.

R. Singh, M. Sahani, and A. Gretton. Kernel instrumental variable regression. In *Advances in Neural Information Processing Systems 32*, pages 4593–4605. 2019.

J. H. Stock and F. Trebbi. Retrospectives: Who invented instrumental variable regression? *Journal of Economic Perspectives*, 17(3):177–194, 2003.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2018.

C. Voloshin, H. M. Le, N. Jiang, and Y. Yue. Empirical study of off-policy policy evaluation for reinforcement learning. *arXiv preprint arXiv:1911.06854*, 2019.

P. Wright. *The Tariff on Animal and Vegetable Oils*. Investigations in International Commercial Policies. Macmillan Company, 1928.