Instrumental Variable Representation Learning under Confounded Covariates

Jungsoo Kim[1] **Kwonho Kim**[1] **Inwoo Hwang**[2] **Sanghack Lee**[1]*

[1] Seoul National University [2] Columbia University

mephisto0525@snu.ac.kr rlarnjsgh99@snu.ac.kr

ih2455@columbia.edu sanghack@snu.ac.kr

Abstract

Instrumental variable (IV) analysis is a crucial tool for causal inference across diverse domains—from genetics to chemistry—in the presence of unobserved confounders, but discovering true IVs from observed covariates is challenging. Recent approaches have focused on synthesizing representations that can serve as IVs, but under restrictive assumptions and settings. We propose CoCoIV to tackle a more challenging yet realistic problem of learning IV representations from observed covariates, potentially correlated with unobserved confounders. CoCoIV utilizes latent variable models to learn representations for both IVs and non-IVs from confounded covariates, guided by a dual prediction network with mutual information regularization, allowing both discrete and continuous treatments. Extensive experiments across various configurations of estimators and treatment types show the effectiveness and wide applicability of our framework.

1 Introduction

When randomized experiments are infeasible, instrumental variable (IV) analysis provides a principled way to address unobserved confounding, widely applied in natural sciences including biology and chemistry [Katan, 1986, Von Hinke et al., 2016, Rajput and Gupta, 2020, Ludl and Michoel, 2021]. An IV influences the treatment X but affects the outcome Y only through X and remains independent of unobserved confounders. However, identifying valid IVs is demanding, especially from high-dimensional data and it is impossible to test the validity conditions without strong assumptions.

Recent work in IV representation learning aims to address this by synthesizing representations that serve as IVs rather than selecting explicit variables [Burgess and Thompson, 2013, Burgess et al., 2016, Kuang et al., 2020, Yuan et al., 2022, Cheng et al., 2023b,a, Wu et al., 2023, Li and Yao, 2024]. These, however, often rely on restrictive assumptions: they assume covariates are independent of unobserved confounders, or they support only binary treatment settings. Such assumptions rarely hold in practice, limiting the reliability of these methods in real-world applications (See Table 1).

Table 1: Comparison: IV representation methods.

Method	Treatment (X) type	Unknown IV	Confounded Covariates
UAS (2013)	Both	✗	✗
WAS (2016)	Both	✗	✗
Ivy (2020)	Discrete-only (+Discrete Y)	✓	✓
AutoIV (2022)	Both	✓	✗
CIV.VAE (2023)	Discrete-only	✓	✗
DVAE.CIV (2023)	Discrete-only	✓	✗
VIV (2024)	Both	✓	✗
CoCoIV (Ours)	Both	✓	✓

In this paper, we tackle a more realistic yet underexplored scenario: learning IV representations when observed covariates themselves potentially are correlated with unobserved confounders (i.e. confounded covariates). We introduce CoCoIV (Instrumental Variable Representation Learning with

*Corresponding author

Accepted to the NeurIPS 2025 Workshop on CauScien: Uncovering Causality in Science.

Confounded Covariates), a framework that uses latent variable modeling to learn distinguishable representations from confounded covariates, each for IV and non-IV. The key components are a dual prediction network and regularization of mutual information between the representations, enforcing they encapsulate information regarding IVs and non-IVs, respectively. Importantly, our method handles both discrete and continuous treatments, highlighting its broad applicability.

We summarize our contributions as follows: (i) To the best of our knowledge, we propose the first framework for IV representation learning under confounded covariates, a challenging yet common scenario in practice; (ii) We introduce a dual prediction network with mutual information regularization, emulating estimation process of widely-accepted IV estimators during the training; (iii) Extensive experimental results demonstrate the effectiveness of our method on generating IV representations from confounded covariates, leading to more reliable causal effect estimation across a wide range of scenarios.

2 Methods

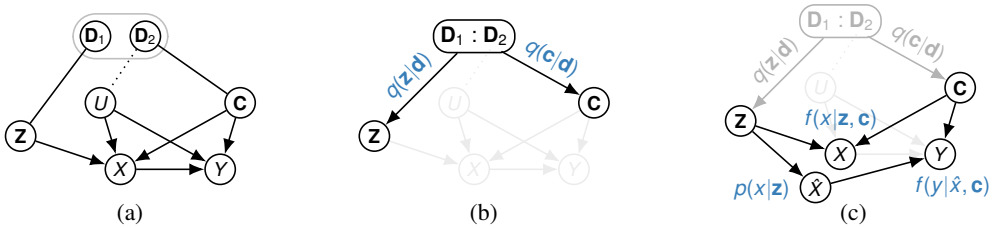


Figure 1: Illustration of our framework. (a) True data generating process. The dotted lines represent the potential correlation between $\mathbf{D}_2$ and U. (b) Inference of $\mathbf{Z}$ and $\mathbf{C}$. (c) Prediction of X and Y.

Problem Setup. We consider a dataset $\{\mathbf{D}^{(i)}, X^{(i)}, Y^{(i)}\}_{i=1}^n$, where $\mathbf{D} \in \mathbb{R}^p$ denotes a set of pre-treatment covariates, X the treatment (binary or continuous), and Y the outcome. Valid IVs are not directly observed, but are assumed to be embedded in $\mathbf{D}$. We assume D can be partitioned into two latent subsets: $\mathbf{D}_1$ containing valid IVs, and $\mathbf{D}_2$ containing non-IVs. This partition is unknown and our goal is to infer IV representation $\mathbf{Z}$ from $\mathbf{D}$ that can be used with usual IV estimators (e.g. 2SLS [Angrist et al., 1996], KernelIV [Singh et al., 2019]) to estimate the causal effect of X on Y.

The causal structure is illustrated in Figure 1a: $\mathbf{D}_1$ and $\mathbf{Z}$ influence X; $\mathbf{D}_2$ and $\mathbf{C}$ affect both X and Y, with possible correlation to unobserved confounders U. This setup extends prior works by explicitly allowing confounded covariates, a setting rarely considered in early IV representation learning methods.

2.1 Overall Framework and Uniqueness

Our proposed framework, CoCoIV, learns representations of $\mathbf{Z}$ (IVs) and $\mathbf{C}$ (non-IVs) from the dataset $\{\mathbf{D}^{(i)}, X^{(i)}, Y^{(i)}\}_{i=1}^n$, composed of three components.

Two encoders ($q(\mathbf{z} \mid \mathbf{d}), q(\mathbf{c}|\mathbf{d})$) **and shared decoder**($p(\mathbf{d}|\mathbf{z}, \mathbf{c})$) are trained to reconstruct $\mathbf{D}$, yielding candidate IV ($\mathbf{Z}$) and non-IV ($\mathbf{C}$) representations (See Figure 1b). After the whole training, we sample IV representation $\mathbf{Z} \sim q(\mathbf{z} \mid \mathbf{d})$ to obtain $\{\mathbf{Z}^{(i)}, X^{(i)}, Y^{(i)}\}_{i=1}^n$, ready for estimating causal effect.

Dual prediction networks (Figure 1c) for treatment X ($f(x|\mathbf{z}, \mathbf{c}), p(x|\mathbf{z})$) are designed to bypass the confounding bias when estimating Y during the training. Unlike other IV representation methods using a single prediction network for X, our method uses $\hat{x}$ derived solely from $\mathbf{z}$ for predicting Y, i.e., $f(y|\hat{x}, \mathbf{c})$ where $\hat{x} \sim p(x|\mathbf{z})$.

Mutual information (MI) regularization that minimizes dependence between $\mathbf{Z}$ and $\mathbf{C}$, encourages the model to disentangle $\mathbf{Z}$ and $\mathbf{C}$ respectively from $\mathbf{D}_1$ and $\mathbf{D}_2$ with the prediction networks.

Our unique component is dual prediction networks. Other IV representation methods with a single prediction network for X, predict Y using X inferred from both $\mathbf{Z}$ and $\mathbf{C}$. As $\mathbf{C}$ is learned to represent confounded covariates, this may inadvertently include confounding effects from U. In contrast, our

method leads to the estimation of causal effect robust to the influence of confounding, which can be viewed as *mimicking* IV estimators. Moreover, CoCoIV supports both binary and continuous treatments, extending applicability beyond prior methods limited to specific data types.

2.2 Learning Objective

The training objective integrates three terms.

ELBO loss $\mathcal{L}_{\text{ELBO}}$ ensures $\mathbf{D}$ can be reconstructed from IV and non-IV representation $\mathbf{Z}, \mathbf{C}$.

$$\mathcal{L}_{\text{ELBO}}, = -\mathbb{E}_q\big[\log p(\mathbf{d} \mid \mathbf{z}, \mathbf{c})] + \int_{\mathbf{c}} q(\mathbf{c} \mid \mathbf{d}) D_{\text{KL}}(q(\mathbf{z} \mid \mathbf{d}) || p(\mathbf{z})) \mathrm{d}\mathbf{c} + \int_{\mathbf{z}} q(\mathbf{z} \mid \mathbf{d}) D_{\text{KL}}(q(\mathbf{c} \mid \mathbf{d}) || p(\mathbf{c})) \mathrm{d}\mathbf{z}$$

Prediction loss $\mathcal{L}_{\text{pred}}$ is designed to learn two separate prediction networks for X, i.e., $p(x \mid \mathbf{z})$ and $f(x \mid \mathbf{z}, \mathbf{c})$, and a prediction network for Y, i.e., $f(y|\hat{x}, \mathbf{c})$.

$$\mathcal{L}_{\text{pred}} = -\mathbb{E}_{q(\mathbf{z},\mathbf{c}|\mathbf{d})}\Big[\log p(x \mid \mathbf{z}) + \text{MSE}(f(x \mid \mathbf{z}, \mathbf{c}), x) + \text{MSE}(f(y \mid \hat{x}, \mathbf{c}), y)\Big],$$

Mutual information loss $\mathcal{L}_{\text{MI}}$ minimizes statistical dependence between $\mathbf{Z}$ and $\mathbf{C}$, using tractable approximations under Gaussian priors.

$$\mathcal{L}_{\text{MI}} = I(\mathbf{Z}; \mathbf{C} \mid \mathbf{D}) = -\frac{1}{2} \sum_{z_j, c_k} \log(1 - Q^2_{z_j, c_k}),$$

The final objective is a weighted sum with balancing coefficients α and β.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ELBO}} + \alpha \cdot \mathcal{L}_{\text{pred}} + \beta \cdot \mathcal{L}_{\text{MI}},$$

3 Empirical Evaluation

We evaluate CoCoIV on both synthetic and real-world datasets, comparing computed treatment effect estimates against baseline IV representation learning methods (UAS [Burgess and Thompson, 2013], WAS [Burgess et al., 2016],DVAE.CIV [Cheng et al., 2023a], AutoIV [Yuan et al., 2022])[2] by applying IV representations from each method to well-known IV estimators (2SLS [Angrist et al., 1996], Ortho [Syrgkanis et al., 2019], KernelIV [Singh et al., 2019]).

Our experiments aim to answer three questions: (i) Can CoCoIV learn reliable IV representations under confounded covariates so that it elicits exact estimate of causal effect? (ii) How does it perform across binary vs. continuous treatments and linear vs. non-linear response functions? (iii) What is the contribution of key component for quality of learned IV representation? Please refer to Appendix B for the details of (i), (ii) with a broader set of estimators and Appendix. C.1 for (iii).

3.1 Synthetic datasets

We conduct experiments with both low-dimensional synthetic data and high-dimensional dataset based on MNIST. Settings vary across binary vs. continuous treatments and linear vs. non-linear response functions. Performance is measured by mean absolute error (MAE) between estimated and true causal effects, while for continuous treatment, we compare MAE between estimated and true potential outcomes. We report MAE with its standard deviation within parentheses in Table 2. For reference, we did not report standard deviations for UAS and WAS since they do not involve *learning*.

As summarized in Table 2, CoCoIV yields more accurate estimates of causal effects for most of the IV-based estimators. In the low-dimensional settings with linear response function, our model tends to attain clear gains when combined with 2SLS and Ortho, while demonstrating modest MAE with KernelIV, which is designed to capture non-linear relationships. In contrast, for non-linear response function, our model obtains the best MAE with KernelIV, owing to its effectiveness in capturing non-linear relationships.

[2]We chose the models where we can access official implementation codes.

Table 2: Compact results (MAE) with different baselines per setting.

		Low-dim						**High-dim**					
		Linear			Nonlinear			Linear			Nonlinear		
		2SLS	Ortho	KernelIV	2SLS	Ortho	KernelIV	2SLS	Ortho	KernelIV	2SLS	Ortho	KernelIV
Bin.	UAS	2.96	3.92	0.22	1.63	3.92	0.58	0.52	17.17	0.22	4.06	6.78	2.41
	WAS	2.25	2.12	**0.21**	2.78	2.12	0.64	0.37	1.74	**0.13**	2.74	4.41	2.10
	DVAE.CIV	1.43 (0.24)	1.99 (1.74)	1.24 (0.02)	**0.83** (0.67)	1.17 (0.71)	0.79 (0.01)	1.87 (1.72)	3.75 (8.24)	0.66 (0.13)	9.31 (28.3)	14.2 (32.48)	**0.70** (0.1)
	CoCoIV	**0.26** (0.09)	**1.3** (1.25)	0.38 (0.31)	1.12 (0.17)	**0.37** (0.47)	**0.33** (0.02)	**0.07** (0.03)	**0.54** (0.44)	0.24 (0.2)	**0.76** (0.06)	**3.11** (0.42)	1.68 (0.29)
Cont.	UAS	1.75	0.5	5.85	2.41	**1.34**	4.87	0.52	17.17	0.22	4.06	6.76	2.41
	WAS	0.79	0.24	3.51	1.57	1.51	5.18	0.37	1.74	**0.13**	2.74	4.4	2.10
	AutoIV	4.15 (9.41)	0.16 (0.2)	**0.7** (0.79)	25.25 (63.06)	1.67 (0.45)	4.26 (1.33)	0.62 (0.76)	1.71 (0.96)	0.32 (0.25)	9.84 (32.47)	2.11 (0.59)	0.96 (0.34)
	CoCoIV	**0.07** (0.09)	**0.02** (0.01)	3.51 (0.55)	**1.38** (0.1)	1.37 (0.1)	**3.82** (1.16)	**0.05** (0.01)	**0.1** (0.08)	0.26 (0.12)	**0.60** (0.05)	**1.89** (0.12)	**0.82** (0.12)

Table 3: Estimates of causal effect on two real-world datasets.

	401(k) (Bin. treatment)			**Police Force (Cont. treatment)**	
	2SLS	Ortho	KernelIV	2SLS	Ortho
UAS	2.23	2.26	0.24	6.96	6.99
WAS	1.68	1.69	0.24	6.96	6.99
DVAE.CIV	0.01 (0.18)	-0.17 (0.71)	0.24 (0.00)	–	–
AutoIV	–	–	–	N/A	N/A
CoCoIV	0.72 (0.08)	1.14 (0.64)	0.24 (0.00)	6.91 (0.07)	6.59 (1.23)

In the high-dimensional settings with **linear** response function, similar to those with low-dimensional datasets, CoCoIV demonstrates the best performance when estimated with 2SLS and Ortho.They achieve the lowest MAE with the smallest standard deviation. Note that simple schemes such as UAS/WAS can appear competitive in certain cases owing to the pixel-level sparsity of MNIST-like covariates, where averaging methods of UAS/WAS effectively down-weights irrelevant zero-pixels. In **continuous** treatment with non-linear response function, which is the most complex settings, our model records the lowest MAEs among all the estimators. These results confirm that the CoCoIV learns IV representation effectively when facing with complex high-dimensional datasets.

3.2 Real-world datasets

We test on two empirical datasets summarized in Table 3 reporting estimated effects. Although known IVs exist, we ignore this information and our model takes the full set of covariates as input, showing that CoCoIV still recovers plausible estimates consistent with prior studies.

For the **401(k)** dataset of Abadie [2003], where the true effect on IRA participation is small and positive (0.03–0.07), our method returns values below 1 and avoids the sign reversals observed in DVAE.CIV. On the **police force** dataset of Chalfin et al. [2022], where the documented effect of police size on quality-of-life arrests is ≈ 5.03, CoCoIV yields estimates in the range 6.5–7.5, closer to the reported effect, while AutoIV produces unstable and explosive values[3].

3.3 Detailed analysis

We investigate the contributions of our key component and quality of learned $\mathbf{Z}$ (See Appendix. C.1).

Dual prediction network and MI regularization Prior approaches often use both IV and non-IV representations to predict X. Our dual design separates $p(x|\mathbf{z})$ and $f(x|\mathbf{z}, \mathbf{c})$, using only the IV-based $\hat{x}$ for modeling Y, reducing estimation bias by 50% on average compared to a single prediction network. Furthermore partial derivative sensitivity analysis shows that with MI ($\beta > 0$), the encoder $q(\mathbf{z}|\mathbf{d})$ focuses more on true IVs $\mathbf{D}_1$, confirming its role in valid IV representation learning.

Quality of learned Z. It is critical to validate whether the learned IV representation $\mathbf{Z}$ satisfies the IV conditions. Mutual information analysis provides an indirect validation, showing that the learned $\mathbf{Z}$ overall has weak association with $\mathbf{C}$ and $\mathbf{D}_2$ (correlated with unobserved confounders) but strong association with genuine IVs $\mathbf{D}_1$.

[3]KernelIV is not applicable as the researchers utilize estimators assuming a linear model.

4 Conclusion

In this work, we tackled the under-explored problem of learning IV representations directly from *confounded covariates*, as outlined in Table 1. By modeling two distinct latent variables and introducing novel components such as a dual prediction network, our method learns representations for both IVs and non-IVs. A key contribution of our approach is its flexible compatibility with diverse data types, enabling robust causal effect estimation across various estimators, treatment types, and functional relationships. Given its effectiveness in high-dimensional datasets, we expect that our method would be applicable to complex observational data in natural science domains, such as climate science datasets, where confounding is pervasive and reliable IV constuction remains a challenge.

References

Alberto Abadie. Semiparametric instrumental variable estimation of treatment response models. *Journal of econometrics*, 113(2):231–263, 2003.

Joshua D. Angrist, Guido W. Imbens, and Donald B. Rubin. Identification of Causal Effects Using Instrumental Variables. *Journal of the American Statistical Association*, 91(434):444–455, 1996.

Joshua David Angrist and Jörn-Steffen Pischke. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton University Press, Princeton, New Jersey, 2009. ISBN 978-0-691-12034-8.

Stephen Burgess and Simon G Thompson. Use of allele scores as instrumental variables for mendelian randomization. *International journal of epidemiology*, 42(4):1134–1144, 2013.

Stephen Burgess, Frank Dudbridge, and Simon G Thompson. Combining information on multiple instrumental variables in mendelian randomization: comparison of allele score and summarized data methods. *Statistics in medicine*, 35(11):1880–1906, 2016.

A Colin Cameron and Pravin K Trivedi. *Microeconometrics: methods and applications*. Cambridge university press, 2005.

Aaron Chalfin, Benjamin Hansen, Emily K Weisburst, and Morgan C Williams Jr. Police force size and civilian race. *American Economic Review: Insights*, 4(2):139–158, 2022.

Debo Cheng, Ziqi Xu, Jiuyong Li, Lin Liu, Thuc Duy Le, and Jixue Liu. Learning conditional instrumental variable representation for causal effect estimation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 525–540. Springer, 2023a.

Debo Cheng, Ziqi Xu, Jiuyong Li, Lin Liu, Jixue Liu, and Thuc Duy Le. Causal inference with conditional instruments using deep generative models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7122–7130, 2023b.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018.

Izrail Moiseevich Gelfand and AM Yaglom. *Calculation of the amount of information about a random function contained in another such function*. American Mathematical Society Providence, 1959.

Lars Peter Hansen. Large sample properties of generalized method of moments estimators. *Econometrica: Journal of the econometric society*, pages 1029–1054, 1982.

Aapo Hyvarinen, Hiroaki Sasaki, and Richard E. Turner. Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning. *arXiv preprint arXiv:1805.08651*, 2019.

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: an unsupervised and generative approach to clustering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1965–1972, 2017.

MartijnB Katan. Apoupoprotein E isoforms, serum cholesterol, and cancer. *The Lancet*, 327(8479): 507–508, 1986.

Yuta Kawakami, Manabu Kuroki, and Jin Tian. Instrumental variable estimation of average partial causal effects. In *International Conference on Machine Learning*, pages 16097–16130. PMLR, 2023.

Ilyes Khemakhem, Diederik P. Kingma, Ricardo Pio Monti, and Aapo Hyvärinen. Variational Autoencoders and Nonlinear ICA: A Unifying Framework. *arXiv preprint arXiv:1907.04809*, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Bohdan Kivva, Goutham Rajendran, Pradeep Ravikumar, and Bryon Aragam. Identifiability of deep generative models without auxiliary information. *Advances in Neural Information Processing Systems*, 35:15687–15701, 2022.

Zhaobin Kuang, Frederic Sala, Nimit Sohoni, Sen Wu, Aldo Córdova-Palomera, Jared Dunnmon, James Priest, and Christopher Re. Ivy: Instrumental Variable Synthesis for Causal Inference. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 398–410. PMLR, 2020.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Xinshu Li and Lina Yao. Distribution-conditioned adversarial variational autoencoder for valid instrumental variable generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13664–13672, 2024.

Manuel Lopez-Martin, Belen Carro, Antonio Sanchez-Esguevillas, and Jaime Lloret. Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in iot. *Sensors*, 17(9):1967, 2017.

Adriaan-Alexander Ludl and Tom Michoel. Comparison between instrumental variable and mediation-based methods for reconstructing causal gene networks in yeast. *Molecular omics*, 17(2):241–251, 2021.

Whitney K Newey. Nonparametric instrumental variables estimation. *American Economic Review*, 103(3):550–556, 2013.

Whitney K. Newey and James L. Powell. Instrumental Variable Estimation of Nonparametric Models. *Econometrica*, 71(5):1565–1578, 2003.

Jaime Pizarroso, José Portela, and Antonio Muñoz. Neuralsens: Sensitivity analysis of neural networks. *Journal of Statistical Software*, 102:1–36, 2022.

Prashant Rajput and Tarun Gupta. Instrumental variable analysis in atmospheric and aerosol chemistry. *Frontiers in Environmental Science*, 8:566136, 2020.

Bhaven Sampat and Heidi L Williams. How do patents affect follow-on innovation? evidence from the human genome. *American Economic Review*, 109(1):203–236, 2019.

Rahul Singh, Maneesh Sahani, and Arthur Gretton. Kernel instrumental variable regression. *Advances in Neural Information Processing Systems*, 32, 2019.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Pierre Stock and Rémi Gribonval. An embedding of relu networks and an analysis of their identifiability. *Constructive Approximation*, 57(2):853–899, 2023.

Vasilis Syrgkanis, Victor Lei, Miruna Oprescu, Maggie Hei, Keith Battocchi, and Greg Lewis. Machine Learning Estimation of Heterogeneous Treatment Effects with Instruments. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Joseph V Terza, Anirban Basu, and Paul J Rathouz. Two-stage residual inclusion estimation: addressing endogeneity in health econometric modeling. *Journal of health economics*, 27(3): 531–543, 2008.

Stephanie Von Hinke, George Davey Smith, Debbie A Lawlor, Carol Propper, and Frank Windmeijer. Genetic markers as instrumental variables. *Journal of Health Economics*, 45:131–148, 2016.

Wing Hung Wong. An equation for the identification of average causal effect in nonlinear models. *Statistica Sinica*, 32, 2022.

Anpeng Wu, Kun Kuang, Ruoxuan Xiong, Minqin Zhu, Yuxuan Liu, Bo Li, Furui Liu, Zhihua Wang, and Fei Wu. Learning instrumental variable from data fusion for treatment effect estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10324–10332, 2023.

Junkun Yuan, Anpeng Wu, Kun Kuang, Bo Li, Runze Wu, Fei Wu, and Lanfen Lin. Auto IV: Counterfactual Prediction via Automatic Instrumental Variable Decomposition. *ACM Trans. Knowl. Discov. Data*, 16(4):1–20, 2022.

Weijia Zhang, Lin Liu, and Jiuyong Li. Treatment effect estimation with disentangled latent factors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10923–10930, 2021.

A Appendix for Method

For reference, Appendix consists of Appendix. A (for **method**), Appendix. B (for **experimental settings** and full results of main experiments), and Appendix. C (for **additional experiments**).

A.1 Formal definition of IV and Real-world example on our DGP

Definition of IV. We define instrumental variables $\mathbf{Z}$ in the relationships with treatment X, outcome Y and unobserved confounders $\mathbf{U}$ (adopted from Angrist and Pischke [2009]).

Definition 1 (Instrumental Variables)**.** a variable Z is called an *instrumental variable (IV)* if three conditions below hold:

1. $Z \not\perp\!\!\!\perp X$ (relevance),
2. $Z \perp\!\!\!\perp Y \mid X, \mathbf{U}$ (exclusion),
3. $Z \perp\!\!\!\perp \mathbf{U}$ (unconfoundedness).

We say that Z is a (valid) IV if all three conditions hold; otherwise, it is a non-IV.

Real-world example. We would like to refer to an interesting example of the study on whether granting a patent leads to follow-on innovations [Sampat and Williams, 2019]. Here, it is infeasible to randomly assign patent acceptance, and further, unobserved confounders (e.g., the innate originality of patent candidates) would distort causality between the treatment and the outcome. The authors used as an IV the assignment of patent examiners, which is nearly random, and its impact rarely directly affects follow-on innovations.

Recall the setting described in Figure. 1a. The assignment of patent examiners ($\mathbf{D}_1$) and their latent technological preferences ($\mathbf{Z}$) influence the treatment (X). On the other hand, research records ($\mathbf{D}_2$) that can be represented by patent-related research trend ($\mathbf{C}$), affect both treatment and outcome, follow-on research (Y) and some of them could be correlated with the era's latent investment trends (U).

A.2 Details on the Model Components

We will introduce details on learning components of the model, especially on the pair of posterior $q(\mathbf{z} \mid \mathbf{d})$ and prior $p(\mathbf{z}) \approx p(\mathbf{z} \mid \mathbf{d})$, and of posterior $q(\mathbf{c}, e_{\mathbf{c}} \mid \mathbf{d})$ and prior $p(\mathbf{c}, e_{\mathbf{c}})$. Then, we will introduce the derivation of our ELBO objective. Finally, model components regarding prediction loss and mutual information regularization will be elaborated.

Modeling regarding Z. we employ different architectures for each encoder, considering the data generating process of $\mathbf{Z}$ and $\mathbf{C}$ in realistic settings where non-IVs are more prevalent and exhibit a more complex structure. We approximate distribution of latent variable $\mathbf{Z}$ through VAE-based framework [Kingma and Welling, 2013, Sohn et al., 2015] considering association between $\mathbf{Z}$ and $\mathbf{D}_1$. We model a prior distribution of $\mathbf{Z}$ as $p(\mathbf{z} \mid \mathbf{d})$; for implementation, we chose a normal distribution as a prior following practice [Sohn et al., 2015, Lopez-Martin et al., 2017]:

$$p(\mathbf{z} \mid \mathbf{d}) = N(\mu_{\mathbf{d}}, \sigma^2_{\mathbf{d}} I).$$

We use $p(\mathbf{z}) = \sum_{\mathbf{d}} p(\mathbf{z}|\mathbf{d})p(\mathbf{d}) \approx \frac{1}{N}\sum_{i=1}^{N} p(\mathbf{z} \mid \mathbf{d}^{(i)})$ where $\mathbf{d}^{(i)} \sim p(\mathbf{d})$. We chose $N = 1$ and intend to learn the prior $p(\mathbf{z}) \approx p(\mathbf{z} \mid \mathbf{d})$.

Then, encoder $q(\mathbf{z} \mid \mathbf{d})$ defined as a variational posterior distribution is trained to be close to $p(\mathbf{z} \mid \mathbf{d})$.

$$q(\mathbf{z} \mid \mathbf{d}) \approx N(\mu_{\mathbf{z}|\mathbf{d}}, \sigma^2_{\mathbf{z}|\mathbf{d}} I).$$

Modeling regarding C. we aim to encode $\mathbf{C}$ that can capture likely complex structure of non-IV $\mathbf{D}_2$ which usually have a number more than that of true IVs in observed covariates in real-world scenarios. By accommodating the mixture model prior, we aim to learn $\mathbf{C}$ expressing complex latent structures of the observed data. Thus, we utilize Variational Deep Embedding (VaDE) [Jiang et al.,

2017], which assumes prior $p(\mathbf{c})$ following a Gaussian Mixture Model. We derive both $\mathbf{c}$ and its component $e_{\mathbf{c}}$ as

$$p(e_{\mathbf{c}}) \sim \mathrm{Cat}(\boldsymbol{\pi})$$
$$p(\mathbf{c} \mid e_{\mathbf{c}}) \sim N(\mu_{e_{\mathbf{c}}}, \sigma^2_{e_{\mathbf{c}}} I),$$

where $\boldsymbol{\pi} = (\pi_1, ..., \pi_K) \in \mathbb{R}^K$. The Loss term related to ELBO in Eq. 2.2 implies that $q(\mathbf{c}, e_{\mathbf{c}} \mid \mathbf{d})$ is trained to be close to its prior $p(\mathbf{c}, e_{\mathbf{c}})$. As VaDE originally assumes, we model $q(\mathbf{c}, e_{\mathbf{c}} \mid \mathbf{d})$ to be a meanfield distribution that can be factorized into $q(\mathbf{c}, e_{\mathbf{c}} \mid \mathbf{d}) = q(\mathbf{c} \mid \mathbf{d}) q(e_{\mathbf{c}} \mid \mathbf{d})$. Then, we will model $q(\mathbf{c} \mid \mathbf{d})$ as,

$$q(\mathbf{c} \mid \mathbf{d}) = N(\mu_{\mathbf{c}|\mathbf{d}}, \sigma^2_{\mathbf{c}|\mathbf{d}} I).$$

For $q(e_{\mathbf{c}} \mid \mathbf{d})$, we compute it as $p(e_{\mathbf{c}} \mid \mathbf{c})$ following the same logic in Jiang et al. [2017]. This is because, as shown in the following rewritten forms,

$$\begin{aligned}
&\mathcal{L}_{\mathrm{ELBO}} \\
&= -\mathbb{E}_{q(\mathbf{z},\mathbf{c},e_{\mathbf{c}}|\mathbf{d})} \left[\log \frac{p(\mathbf{d}_1, \mathbf{d}_2, \mathbf{z}, \mathbf{c}, e_{\mathbf{c}})}{q(\mathbf{z}, \mathbf{c}, e_{\mathbf{c}} \mid \mathbf{d})} \right] \\
&= -\int_{\mathbf{z},\mathbf{c}} \sum_{e_{\mathbf{c}}} q(\mathbf{z} \mid \mathbf{d}) q(\mathbf{c} \mid \mathbf{d}) q(e_{\mathbf{c}} \mid \mathbf{d}) \\
&\qquad \cdot \left[\log \frac{p(\mathbf{d}_1, \mathbf{z}) p(\mathbf{d}_2 \mid \mathbf{c}) p(\mathbf{c})}{q(\mathbf{z} \mid \mathbf{d}) q(\mathbf{c} \mid \mathbf{d})} + \log \frac{p(e_{\mathbf{c}} \mid \mathbf{c})}{q(e_{\mathbf{c}} \mid \mathbf{d})} \right] \mathrm{d}\mathbf{z}\, \mathrm{d}\mathbf{c} \\
&= -\int_{\mathbf{z},\mathbf{c}} q(\mathbf{z} \mid \mathbf{d}) q(\mathbf{c} \mid \mathbf{d}) \log \frac{p(\mathbf{d}_1, \mathbf{z}) p(\mathbf{d}_2 \mid \mathbf{c}) p(\mathbf{c})}{q(\mathbf{z} \mid \mathbf{d}) q(\mathbf{c} \mid \mathbf{d})} \mathrm{d}\mathbf{z} \mathrm{d}\mathbf{c} \\
&\qquad + \int_{\mathbf{z},\mathbf{c}} q(\mathbf{z} \mid \mathbf{d}) q(\mathbf{c} \mid \mathbf{d}) D_{\mathrm{KL}}(q(e_{\mathbf{c}} \mid \mathbf{d}) \parallel p(e_{\mathbf{c}} \mid \mathbf{c})) \mathrm{d}\mathbf{z}\, \mathrm{d}\mathbf{c}
\end{aligned}$$

to optimize $\mathcal{L}_{\mathrm{ELBO}}$, $D_{\mathrm{KL}}(q(e_{\mathbf{c}} \mid \mathbf{d}) \parallel p(e_{\mathbf{c}} \mid \mathbf{c}))$ should be 0.

Modeling of the decoder $p(\mathbf{d}|\mathbf{z}, \mathbf{c})$. A shared decoder $p(\mathbf{d}|\mathbf{z}, \mathbf{c})$ is trained to have $\widehat{\mathbf{Z}} \sim q(\mathbf{z}|\mathbf{d})$ and $\widehat{\mathbf{C}} \sim q(\mathbf{c}|\mathbf{d})$ obtaining enough dependence with $\mathbf{D}$. As we have two distinct architectures for the encoders, the reconstruction term $\mathbb{E}_q[p(\mathbf{d}|\mathbf{z}, \mathbf{c})]$ can be collapsed in the early stage of the training. Thus, we pretrain the shared decoder and encoders $q(\mathbf{z}|\mathbf{d})$ and $q(\mathbf{c}|\mathbf{d})$ with 30 epochs.

Details on ELBO of $\mathbf{D}$. We can derive the negative ELBO of $\log p(\mathbf{d})$ with the components above.

$$\begin{aligned}
-\log p(\mathbf{d}) &\leq -\mathbb{E}_{q(\mathbf{z},\mathbf{c},e_{\mathbf{c}}|\mathbf{d})} \left[\log \frac{p(\mathbf{d}, \mathbf{z}, \mathbf{c}, e_{\mathbf{c}})}{q(\mathbf{z}, \mathbf{c}, e_{\mathbf{c}} \mid \mathbf{d})} \right] \\
&= -\mathbb{E}_{q(\mathbf{z},\mathbf{c},e_{\mathbf{c}}|\mathbf{d})} \left[\log \frac{p(\mathbf{d} \mid \mathbf{z}, \mathbf{c}) p(\mathbf{z}) p(\mathbf{c} \mid e_{\mathbf{c}}) p(e_{\mathbf{c}})}{q(\mathbf{z}, \mathbf{c}, e_{\mathbf{c}} \mid \mathbf{d})} \right] \\
&= -\mathbb{E}_q \big[\log p(\mathbf{d} \mid \mathbf{z}, \mathbf{c}, e_{\mathbf{c}}) \big] \\
&\qquad + \int_{\mathbf{c}} \sum_{e_{\mathbf{c}}} q(\mathbf{c}, e_{\mathbf{c}} \mid \mathbf{d}) D_{\mathrm{KL}}(q(\mathbf{z} \mid \mathbf{d}) \parallel p(\mathbf{z})) \mathrm{d}\mathbf{c} \\
&\qquad + \int_{\mathbf{z}} q(\mathbf{z} \mid \mathbf{d}) D_{\mathrm{KL}}(q(\mathbf{c}, e_{\mathbf{c}} \mid \mathbf{d}) \parallel p(\mathbf{c}, e_{\mathbf{c}})) \mathrm{d}\mathbf{z} \\
&\approx -\mathbb{E}_q \big[\log p(\mathbf{d} \mid \mathbf{z}, \mathbf{c}) \big] \\
&\qquad + \int_{\mathbf{c}} \sum_{e_{\mathbf{c}}} q(\mathbf{c}, e_{\mathbf{c}} \mid \mathbf{d}) D_{\mathrm{KL}}(q(\mathbf{z} \mid \mathbf{d}) \parallel p(\mathbf{z} \mid \mathbf{d})) \mathrm{d}\mathbf{c} \\
&\qquad + \int_{\mathbf{z}} q(\mathbf{z} \mid \mathbf{d}) D_{\mathrm{KL}}(q(\mathbf{c}, e_{\mathbf{c}} \mid \mathbf{d}) \parallel p(\mathbf{c}, e_{\mathbf{c}})) \mathrm{d}\mathbf{z} \\
&= \widehat{\mathcal{L}_{\mathrm{ELBO}}}.
\end{aligned}$$

As we approximate $p(\mathbf{z})$ with $p(\mathbf{z} \mid \mathbf{d})$, the derived term incorporates the components of the model $p(\mathbf{d} \mid \mathbf{z}, \mathbf{c}), p(\mathbf{z} \mid \mathbf{d}), p(\mathbf{c}, e_c), q(\mathbf{z} \mid \mathbf{d}), q(\mathbf{c}, e_c \mid \mathbf{d})$.

Modeling $p(x \mid \mathbf{z})$. We parameterize it to approximate $p(x \mid \mathbf{z})$ assumed to follow distribution in regular exponential family.

$$p_\theta(x \mid \mathbf{z}) = \begin{cases} \text{Bern}(\text{sigmoid}(\theta_1(\mathbf{z}))) & \text{for binary } x \\ N(\mu_{\theta_2(\mathbf{z})}, \sigma^2_{\theta_2(\mathbf{z})}). & \text{for continuous } x \end{cases}$$

One can further model it with more complex exponential family, but for simplicity we select the two distributions each of which covers either binary or continuous variable.

Tractable MI regularization. Since we model both $q(\mathbf{z} \mid \mathbf{d})$ and $q(\mathbf{c} \mid \mathbf{d})$ as normal distributions, we can obtain tractable MI as follows [Gelfand and Yaglom, 1959]:

$$I(Z_j; C_k|\mathbf{D}) = -\frac{1}{2}\log(1 - Q^2_{z_j,c_k}),$$

for all $z_j \in \mathbf{Z}$ and $c_k \in \mathbf{C}$, where $Q_{z_j,c_k} = \text{corr}(z_j, c_k)$ can be replaced with the sample correlation $\hat{Q}_{z_j,c_k}$ from mini- batch. Thus, we can formulate the training objective in terms of MI as follows:

$$\mathcal{L}_3 = -\frac{1}{2}\sum_{j,k}\log(1 - \hat{Q}^2_{z_j,c_k}). \tag{1}$$

Instead, we use a simplified version of the formula, $\mathcal{L}_{3,\text{simple}} = \sum_{j,k}\hat{Q}^2_{z_j,c_k}$, which has the same global optimum when $\hat{Q}_{z_j,c_k} = 0$ for all z_j and c_k, but does not explode.

A.3 Discussion on identification of latent variable models

In general, identifiability of latent variable models often requires auxiliary variables or additional conditions [Hyvarinen et al., 2019, Khemakhem et al., 2020].

Here, we provide conditions for the model identifiability and how we considered them in our architectural design:

- When the prior distribution of latent variable $\mathbf{Z}, \mathbf{C}$ follows a Gaussian Mixture Model (possibly, a number of components can be one or more);
- When $\mathbf{D}_1, \mathbf{D}_2$ are fed into respectively learned encoders; and
- When the function that maps latent variables to observed variables is injective;

It is known that the model is identifiable up to affine transformations without auxiliary variables if the above conditions are satisfied [Kivva et al., 2022]. As such conditions are often too strict and infeasible, prior works [Zhang et al., 2021, Yuan et al., 2022, Cheng et al., 2023b,a, Li and Yao, 2024] also lack the guarantee of the model identifiability. In our work, we employed ReLU activations [Stock and Gribonval, 2023] and MI constraints to impose them in practice where its effectiveness is demonstrated in Sec. 3, Appendix. C.

A.4 Discussion on identification of causal effect

If the IV representation model recovers $\mathbf{Z}$ in underlying graphical model Fig. 1a, causal effect of X on Y is identified by following the existing works of IV analysis. The conditions for the identification can be listed as follows; the influence of the treatment on the outcome is clearly separable from those of confounders [Terza et al., 2008, Kawakami et al., 2023]; specifically, when the relationships among variables are linear, it is sufficient that the dimension of $\mathbf{Z}$ is more than that of X, which is usually called rank condition [Cameron and Trivedi, 2005, Newey, 2013]. For a non-parametric setting, the completeness of a set of distributions conditioned on $\mathbf{Z}$ (i.e., $X|Z$) should be considered. [Newey and Powell, 2003, Wong, 2022].

Although these conditions vary among the different IV estimators, we tried our best to reflect the conditions when generating synthetic datasets for empirical validations.

B Appendix for Experiment

Here, we report the full results on synthetic datasets with more IV estimators in Table 4, 5. The results involve three more IV estimators (IVGMM, DML, Poly2SLS). The details of dataset and experiment settings are described below.

B.1 Dataset Details

Synthetic Datasets We create synthetic datasets with sample sizes of 5000. In order to make the setting more realistic, in both low-dimensional and high-dimensional cases, approximately the half of the variables in $\mathbf{D}_2$ are correlated with an unobserved confounder denoted as U. Additionally, the number of instrumental variables (IVs) in $\mathbf{D}$ is less than the half of all the observed covariates.

Datasets are generated based on the following process. The dimensions of $\mathbf{D}_1, \mathbf{Z}, \mathbf{D}_2, \mathbf{C}$ are denoted as m, k, l, o, respectively. Distributions of exogenous variables and variables without parents is $\epsilon_{\mathbf{D}_1} \sim N(0, 10I_m)$, $\epsilon_{\mathbf{Z}} \sim N(0, 0.01I_k)$, $\epsilon_{\mathbf{D}_2} \sim N(0, 0.25I_l)$, $U \sim N(0, 1)$, and $\mathbf{C} \sim N(0, I_o)$. The rest are determined as

$$
\begin{aligned}
&\mathbf{D}_1 = f_{\mathbf{D}_1}(\epsilon_{\mathbf{D}_1}), \mathbf{Z} = f_{\mathbf{Z}}(\mathbf{D}_1) + \epsilon_{\mathbf{Z}}, \\
&\mathbf{D}_2 = A^\top \mathbf{C} + \epsilon_{\mathbf{D}_2} + U \cdot 1(\mathbf{p} > 0.5), \\
&\qquad \text{where, } p_i \sim \text{Bern}(0.5),\ i = 1, 2, \ldots, l, \\
&X = \begin{cases} 1\left[\dfrac{1}{1 + \exp(f_X(\mathbf{Z}) + B^\top[\mathbf{C} : \mathbf{D}_2] + U)} > 0.5\right] & \text{(binary)} \\ f_X(\mathbf{Z}) + B^\top[\mathbf{C} : \mathbf{D}_2] + U & \text{(continuous)} \end{cases} \\
&Y = g(X) + f_Y(\mathbf{C}, \mathbf{D}_2) + U,
\end{aligned}
$$

where $[\mathbf{C} : \mathbf{D}_2]$ is the concatenation of $\mathbf{C}$ and $\mathbf{D}_2$.

We apply an affine transformation of $\mathbf{C}$ and $\mathbf{D}_2$ by using coefficient matrices A and B sampled from a normal distribution, respectively. Other arbitrary functional relationships between variables are introduced by sampling $f_{(.)} \sim \mathcal{GP}(0, k_{\text{RBF}})$, i.e., a Gaussian Process prior with zero mean function and RBF kernel with its length scale set to 1. The function sampler is intended to incorporate the complexity of real-world scenarios.

We generate X and Y using the same process in the low-dimensional and high-dimensional case. For the response function relating X to Y, denoted as $g(X)$, we investigate two specific cases: a linear relationship, where $g(X) = 3X$, and a non-linear relationship, where $g(X) = \exp(0.5X)$. In other words, in the linear case, the true ATE is 3, while in the non-linear case, the true APCE is given by $\partial_x \mathbb{E}[Y(X = x)] = 0.5 \exp(0.5x)$.

In low-dimensional scenarios, each dimension of the variable, i.e., m, k, l, o is set to 2, 2, 4, 3, and in high-dimensional scenarios, we utilize the MNIST dataset [LeCun et al., 1998]. Images in the dataset are represented by $\mathbf{D} \in \mathbb{R}^{28 \times 28 = 784}$, i.e. $m + l = 784$. Due to the lack of specific information about which variables (pixels) are $\mathbf{D}_1$ or $\mathbf{D}_2$, we implicitly assume that $\mathbf{D}_1$ is a set of pixels corresponding to digit. However, in this case, $\mathbf{D}_1$ can be changed by each image, which is more challenging. In terms of assumption about distinguishable $\mathbf{D}$, we can consider it as an even more general setting that should learn the representation of $\mathbf{Z}$ from indistinguishable $\mathbf{D}$. Additionally, we set $\mathbf{Z}$ as the label of each image, thus establishing an intuitive relationship between $\mathbf{D}_1$ and $\mathbf{Z}$. To create distinctive values for $\mathbf{D}_2$, the first 100 pixels are selected and noise U is injected, which can serve as a confounding variable for digit recognition.

For each dataset, we conduct 20 independent replications with the training and test datasets, divided in a 7:3 ratio.

Relationships between Participation in Tax-deferred Programs [Abadie, 2003]. The dataset is composed of 11 variables with 9,275 units. Except for the treatment (participation in 401(k)) and the outcome (participation in IRA), income, net family financial assets, family size, age, gender etc. were collected. Among 11 variables, we use 6 numerical variables that are suitable for averaging methods such as UAS and WAS. As the outcome variable is binary, we extend our model to binary outcome by introducing binary cross entropy to model $f(y \mid \hat{x}, \mathbf{c})$. The dataset is available in R package.[4]

[4] `https://cran.r-project.org/web/packages/wooldridge/wooldridge.pdf`

Table 4: Experimental results on low-dimensional synthetic datasets.

		Linear response function						Non-linear response function					
	Method	2SLS	IVGMM	DML	Ortho	Poly2SLS	KernelIV	2SLS	IVGMM	DML	Ortho	Poly2SLS	KernelIV
Binary	UAS	2.96	2.96	4.99	3.92	1.96	0.22	1.63	1.63	4.63	3.92	1.96	0.58
	WAS	2.25	2.25	3.61	2.12	**1.06**	**0.21**	2.78	2.78	2.92	2.12	1.05	0.64
	DVAE.CIV	1.43 (0.24)	1.43 (0.24)	1.99 (1.74)	1.64 (2.7)	1.12 (0.88)	1.24 (0.02)	**0.83** (0.67)	**0.83** (0.67)	0.88 (0.42)	1.17 (0.71)	1.8 (2.4)	0.79 (0.01)
	CoCoIV	**0.26** (0.09)	**0.31** (0.07)	**1.59** (1.17)	**1.3** (1.25)	1.66 (3.24)	0.38 (0.31)	1.12 (0.17)	1.06 (0.2)	**0.4** (0.56)	**0.37** (0.47)	**0.47** (0.18)	**0.33** (0.02)
Continuous	UAS	1.75	1.75	0.52	0.5	1.4	5.85	2.41	2.41	**1.34**	**1.34**	6.28	4.87
	WAS	0.79	0.79	0.25	0.24	0.69	3.51	1.57	1.57	1.52	1.51	4.95	5.18
	AutoIV	4.15 (9.41)	4.15 (9.41)	0.23 (0.47)	0.16 (0.2)	0.7 (0.79)	**3.18** (0.9)	25.25 (63.06)	25.25 (63.06)	1.65 (0.41)	1.67 (0.45)	4.99 (0.94)	4.26 (1.33)
	CoCoIV	**0.07** (0.09)	**0.08** (0.09)	**0.03** (0.03)	**0.02** (0.01)	**0.35** (0.07)	3.51 (0.55)	**1.38** (0.1)	**0.3** (0.15)	1.35 (0.12)	1.37 (0.1)	**4.13** (0.32)	**3.82** (1.16)

Table 5: Experimental results on high-dimensional synthetic datasets.

		Linear response function						Non-linear response function					
	Method	2SLS	IVGMM	DML	Ortho	Poly2SLS	KernelIV	2SLS	IVGMM	DML	Ortho	Poly2SLS	KernelIV
Binary	UAS	0.52	0.52	N/A	17.17	14.57	0.22	4.06	4.06	7.06	6.78	3.08	2.41
	WAS	0.37	0.37	1.74	1.74	0.85	**0.13**	2.74	2.74	4.35	4.41	2.09	2.1
	DVAE.CIV	1.87 (1.72)	1.87 (1.72)	1.96 (2.05)	3.75 (8.24)	**0.5** (0.13)	0.66 (0.11)	9.31 (28.3)	9.31 (28.3)	**2.48** (4.08)	14.2 (32.48)	**0.75** (0.11)	**0.7** (0.1)
	CoCoIV	**0.07** (0.03)	**0.05** (0.02)	**0.78** (0.8)	**0.54** (0.44)	0.61 (0.49)	0.24 (0.2)	**0.76** (0.06)	**0.7**(0.07)	3.19 (0.5)	**3.11** (0.42)	1.37 (0.19)	1.68 (0.29)
Continuous	UAS	0.52	0.52	16.92	17.1	14.57	0.22	4.06	4.06	6.79	6.76	3.08	2.41
	WAS	0.37	0.37	1.76	1.72	0.85	**0.13**	2.74	2.74	4.43	4.4	2.09	2.1
	AutoIV	0.62 (0.76)	0.62 (0.76)	1.65 (0.94)	1.71 (0.96)	1.19 (0.5)	0.32 (0.25)	9.84 (32.47)	9.84 (32.47)	2.14 (0.59)	2.11 (0.59)	0.93 (0.32)	0.96 (0.34)
	CoCoIV	**0.05** (0.01)	**0.02** (0.01)	**0.23** (0.21)	**0.1** (0.08)	**0.24** (0.06)	0.26 (0.12)	**0.6** (0.05)	**0.6** (0.05)	**2.0** (0.19)	**1.89** (0.12)	**0.85** (0.07)	**0.82** (0.12)

Police Force Size and Civilization Race [Chalfin et al., 2022]. The dataset is composed of 195 columns with 1037 units. Among 195 columns, regardless of methods, we use city id (with one-hot encoding), interacted state by year, power of population, detailed data for city's race, gender and age composition, median household income, poverty rate, and unemployment rate except for the treatment (number of sworn police officers) and outcome (Quality of life arrests for white). We also included IV (variation in the timing of federal block grants provided by the US Department of Justice's Community Oriented Policing Services (COPS) office) in those covariates. The dataset is available on replication package.[5]

B.2 Baselines

UAS. Unweighted Allele Score (UAS) [Burgess and Thompson, 2013] was devised for Mendelian randomization [Katan, 1986]. The model calculates the average of each covariate per unit and uses the value as an IV. The model assumes that all averaged covariates satisfy the conditions of IVs, but they are weak. We implement UAS with basic functions of average provided by *python* packages *pytorch* and *numpy* by following the instructions in the original papers.

WAS. Weighted Allele Score (WAS) [Burgess et al., 2016] takes a similar approach with UAS, but it calculates a weighted average of values of each covariate. The weights are calculated from the correlation between the covariates and treatment. WAS also assumes that all the covariates satisfy the conditions of IVs. We implement WAS in a similar way to UAS.

DVAE.CIV DVAE.CIV [Cheng et al., 2023a] learns a representation of conditional IV and its conditioning set. Conditional IV requires conditioning set to function as IV. DVAE.CIV learns the two representations by leveraging disentangling Variational Autoencoder. For implementation, we use the code provided by the authors.[6]

B.3 Estimators

For IV-based estimators, 2SLS and IVGMM [Hansen, 1982] are implemented with python package *linearmodels*. We use the implementation of DML [Chernozhukov et al., 2018], Ortho [Syrgkanis et al., 2019] in python package *econml*. As Ortho and DML do not provide a prediction method, we use the value of response function when the treatment value is 1 and 0. Default parameters are used for each class. For Poly2SLS and KernelIV [Singh et al., 2019], we used open sourced implementation.

[5] `https://www.aeaweb.org/articles?id=10.1257/aeri.20200792` and `https://www.openicpsr.org/openicpsr/project/135761/version/V1/view`.

[6] As DVAE.CIV is an advanced version of CIV.VAE, we omitted experiments on CIV.VAE.

Table 6: Hyperparameters of the model.

Parameters	Low-dimensional		High-dimensional	
	binary	continuous	binary	continuous
Dim of $\mathbf{Z}$	3	3	25	25
Dim of $\mathbf{C}$	2	2	25	25
Number of components	4	4	4	4
Hidden dim	200	200	150 or 200	150 or 200
Hidden layers	1 or 3	1 or 3	1 or 3	1 or 3
Activation Function	(ReLU, Sigmoid)	(ReLU, Sigmoid)	(ReLU, Sigmoid)	(ReLU, Sigmoid)

Table 7: Hyperparameters for training.

Dim. of $\mathbf{Z}$ (low-dim)	3	Dim. of $\mathbf{C}$ (low-dim)	2
Dim. of $\mathbf{Z}$ (high-dim)	25	Dim. of $\mathbf{C}$ (high-dim)	25
Prediction loss weight α	5	Weight decay	0.0001
MI loss weight β	5	Optimizer	Adam
Batch size	512	Training epochs	128
Learning Rate	0.001	LR Decay	0.01
LR Scheduler	StepLR		

B.4 Implementation

Our method have VaDE architecture to learn complex distributions. We used open-sourced implementation of VaDE.

As VaDE method did, we also pretrain our encoder and decoder to avoid the problem that the reconstruction term would be weak [Jiang et al., 2017]. Additionally, we select Adam optimizer and apply early stopping for efficient learning. All layers are fully connected, including encoders, decoder, and auxiliary nets. Hyperparameter details are on the Table 6 and Table 7 according to the experiment settings with dimensions and type of treatment. In the Table 6, "or" in "Number of components" statement means that we applied both values with linear or non-linear case respectively. "or" in "Hidden layers" means that we chose either one or three layers when constructing networks. On the other hand, in the Table 7, early stopping epochs are applied with 3 when training most of low-dimensional cases. High dimensional case is trained with early stopping epochs 10. We tuned the hyperparameters with grid search within a specific range. The dimensions of latent variable $\mathbf{Z}$ are selected from $[1, 2, 3]$ for low-dimensional setting and $[10, 20, 25]$ for high-dimensional setting. The dimensions of $\mathbf{C}$ are selected from the same range, and the number of its component is selected from $[2, 3, 4]$. Experiments were executed on NVIDIA RTX 6000 with 48GB memory.

C Additional Experimental Results

Full results of the ablation study on various estimators are reported in Table 8. The results indicate the importance of each component in our model. (1) The lack of auxiliary regression networks leads to a significant decline in performance. (2) The dual prediction nets take an important role for the inference of unconfounded representation from the perspective of performance decline. We also provide the results of experiments on various latent model architectures for representation $\mathbf{Z}$ and $\mathbf{C}$, which is elaborated in the last part of the section.

C.1 Detailed Analysis on key components and learned representations

We now provide a detailed analysis of the quality of learned IV representation $\mathbf{Z}$ (Figure 3), the importance of key components of CoCoIV. This part is an extended illustration of Sec. 3.3.

Dual vs. single prediction network for X (Figure 2, Table. 8) Our model intends to mimic usual causal inference with IVs through a dual prediction network $p(x|\mathbf{z}), f(x|\mathbf{z}, \mathbf{c})$, which is one of the key components of the model. To examine its effectiveness, we devise a model using only a single prediction network for treatment X, i.e., $f(x|\mathbf{z}, \mathbf{c})$. This approach, estimating Y with $X \sim f(x|\mathbf{z}, \mathbf{c})$,

Table 8: Experiment results on ablation study.

		Linear response function						Non-linear response function					
	Method	2SLS	IVGMM	DML	Ortho	Poly2SLS	KernelIV	2SLS	IVGMM	DML	Ortho	Poly2SLS	KernelIV
Binary	Ours	0.26 (0.09)	0.31 (0.07)	**1.59** (1.17)	**1.3** (1.25)	**1.66** (3.24)	**0.38** (0.31)	**1.12** (0.17)	**1.06** (0.2)	**0.4** (0.56)	**0.37** (0.47)	**0.47** (0.18)	**0.33** (0.02)
	w/o pred	0.26 (0.08)	0.32 (0.08)	3.33 (1.8)	3.42 (1.47)	2.82 (2.22)	0.42 (0.44)	1.46 (0.11)	1.37 (0.1)	3.78 (1.87)	3.71 (1.52)	34.41 (143.74)	0.5 (0.23)
	w/o aux	**0.23** (0.06)	**0.29** (0.06)	5.67 (6.8)	3.13 (1.71)	2.17 (1.2)	**0.38** (0.4)	1.49 (0.1)	1.41 (0.1)	4.97 (5.25)	3.13 (1.71)	2.12 (1.24)	0.68 (0.3)
Continuous	Ours	**0.07** (0.09)	**0.08** (0.09)	**0.03** (0.03)	**0.02** (0.01)	**0.35** (0.07)	**3.51** (0.55)	**1.38** (0.1)	**0.3** (0.15)	**1.35** (0.12)	**1.37** (0.1)	**4.13** (0.32)	**3.82** (1.16)
	w/o pred	1.42 (1.69)	1.48 (1.69)	0.69 (0.72)	0.35 (0.48)	2.37 (2.75)	4.78 (1.63)	1.42 (0.55)	1.31 (1.99)	1.75 (0.75)	1.54 (0.6)	4.95 (0.77)	6.6 (1.29)
	w/o aux	1.4 (1.17)	1.47 (1.14)	0.92 (1.21)	0.38 (0.24)	1.96 (2.02)	4.96 (1.51)	1.72 (1.67)	2.12 (2.79)	2.52 (2.68)	1.43 (0.77)	5.19 (1.01)	6.4 (1.59)

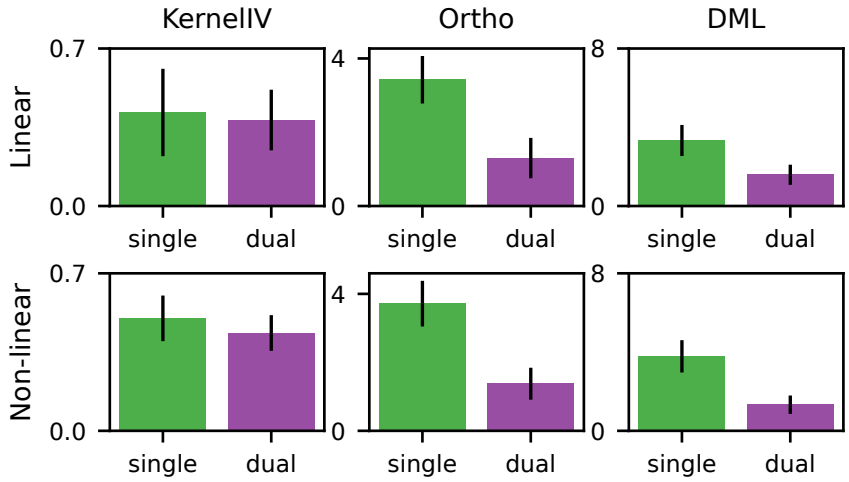


Figure 2: Ablation on the dual prediction network with binary treatment. Green and purple bars indicate MAE from the model with dual and single prediction networks, respectively.

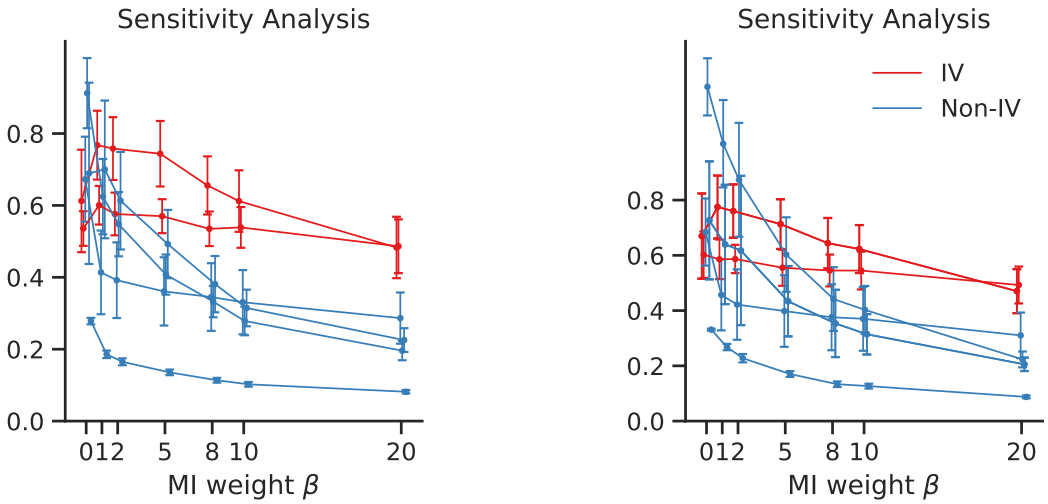
poses a risk of introducing confounding derived from $\mathbf{c}$, which was employed by prior works such as AutoIV and DVAE.CIV. As expected, Figure 2 demonstrates that employing a dual prediction network mitigates bias in causal effect estimates with generated $\mathbf{Z}$.

Mutual Information regularization (Figure 3) Our model aims at learning representations of $\mathbf{Z}$ from unknown IVs in $\mathbf{D}$. In real-world data, we have no access to information on which variable is genuinely valid IV. Thus, to verify our model learning representation of $\mathbf{Z}$ from IVs, we examine how the model is sensitive to observed covariates when encoding $\mathbf{Z}$. We execute sensitivity analysis for each covariate in synthetic data where we know which variable is true IV. Specifically, we measure partial derivative sensitivity Pizarroso et al. [2022] for each $d_l \in \mathbf{D}$, i.e., $\frac{1}{np}\sum_{i=1}^{n}\sum_{j=1}^{p}\left|\frac{\partial z_j^{(i)}}{\partial d_l^{(i)}}\right|$, where n is a number of test data and p is the dimension of $\mathbf{Z}$. A large value of sensitivity indicates that the output highly depends on the changes in the corresponding covariate.

As shown in Figs. 3a and 3b, with MI regularization ($\beta > 0$), the learned encoder for $\mathbf{Z}$ is more sensitive to IVs than to non-IVs. Fig. 3b implies that when $\beta = 0$, i.e., without MI regularization, generated $\mathbf{Z}$ can be highly influenced by non-IVs in $\mathbf{D}_2$. These results indicate that by regularizing MI between the two latent variables sufficiently, encoder $q(\mathbf{z} \mid \mathbf{d})$ may derive $\mathbf{z}$ depending more on valid IVs in $\mathbf{D}_1$.

Quality of learned representation Z (Table. 9) Furthermore, we investigate whether the learned IV representation $\mathbf{Z}$ genuinely satisfy conditions of IV and whether $\mathbf{Z}$ and $\mathbf{C}$ are well disentangled. Specifically, to examine whether $\mathbf{Z}$ is independent of unobserved confounders U (unconfoundedness), we use $\mathbf{D}_2$ (which is partially correlated with U) as a proxy.[7]

[7]We examined the results on low-dimensional datasets, because in high-dimensional datasets, where $\mathbf{D}_1$ refers to the pixels that make up each digit label, we cannot directly access or specify $\mathbf{D}_1$.



(a) Linear response function

(b) Nonlinear response function

Figure 3: Ablation analysis for MI regularization coefficient β in datasets with binary treatment. Red and blue lines are sensitivity for IVs and non-IVs in covariates, respectively.

Table 9: Mutual Information Comparison among Variables.

		Average Mutual Information				
Treatment	Response Function	$\mathbf{Z}$ vs $\mathbf{C}$	$\mathbf{Z}$ vs $\mathbf{D}_1$	$\mathbf{Z}$ vs $\mathbf{D}_2$	$\mathbf{C}$ vs $\mathbf{D}_2$	$\mathbf{C}$ vs $\mathbf{D}_1$
Binary	Linear	0.12 (0.09)	0.26 (0.23)	0.17 (0.17)	0.39 (0.16)	0.12 (0.11)
	Non-linear	0.11 (0.06)	0.25 (0.24)	0.21 (0.18)	0.41 (0.19)	0.13 (0.12)
Continuous	Linear	0.09 (0.03)	0.22 (0.13)	0.04 (0.04)	0.29 (0.16)	0.07 (0.12)
	Non-linear	0.29 (0.12)	0.34 (0.11)	0.05 (0.05)	0.26 (0.1)	0.08 (0.06)

As described in Table. 9, we calculate the average mutual information between $Z_i \in \mathbf{Z}$ and variables in $\mathbf{D}_2$, as well as with components in $\mathbf{C}$.

Results show that while the model struggles in the continuous treatment with nonlinear response—our most complex experimental setting, in general $\mathbf{Z}$ has weak association with both $\mathbf{C}$ and $\mathbf{D}_2$, but stronger association with $\mathbf{D}_1$, indicating that $\mathbf{Z}$ effectively aligns with true IVs.

As shown in Figs. 3a and 3b, with MI regularization ($\beta > 0$), the learned encoder for $\mathbf{Z}$ is more sensitive to IVs than to non-IVs. Figure 3b implies that when $\beta = 0$, i.e., without MI regularization, generated $\mathbf{Z}$ can be highly influenced by non-IVs in $\mathbf{D}_2$. These results indicate that by regularizing MI between the two latent variables sufficiently, encoder $q(\mathbf{z} \mid \mathbf{d})$ may derive $\mathbf{z}$ depending more on valid IVs in $\mathbf{D}_1$.

C.2 Experiments on robustness and architecture choice.

Experiments on assumption violation. We assess how robust our model is to dependence among observed covariates $\mathbf{D}$. This is a violation of assumption $\mathbf{D}_1 \perp\!\!\!\perp \mathbf{D}_2$ in Sec. 2. In this case, as mentioned in Sec. 3.3, learned representation Z may not be a valid IV because the variables in $\mathbf{D}_1$ are not IVs anymore. However, in real-world settings, IVs are seldom perfectly unconfounded but are more often in a corrupted state, influenced by other variables. The main goal of our experiment is thus to determine whether our model could robustly estimate the causal effect even within such compromised circumstances.

We execute experiments on the datasets assuming the existence of unobserved common causes between $\mathbf{D}_1$ and $\mathbf{D}_2$. The generating process is similar with that in Sec. B.1 except for the presence

Table 10: Experiment results on low-dimensional synthetic datasets within dependency.

		Linear response function						Non-linear response function					
	Method	2SLS	IVGMM	DML	Ortho	Poly2SLS	KernelIV	2SLS	IVGMM	DML	Ortho	Poly2SLS	KernelIV
Binary	UAS	3.73	3.73	5.53	3.64	1.75	0.58	2.37	2.37	5.0	3.64	1.77	**0.45**
	WAS	0.52	0.52	2.29	**1.39**	**0.65**	**0.43**	1.35	1.35	1.8	1.39	**0.67**	**0.45**
	DVAE.CIV	1.52 (0.23)	1.52 (0.23)	2.78 (2.27)	3.17 (6.92)	1.14 (0.2)	1.27 (0.03)	**0.76** (0.5)	**0.76** (0.5)	1.49 (1.3)	2.48 (3.32)	0.93 (0.1)	0.94 (0.02)
	CoCoIV(Ours)	**0.33** (0.1)	**0.34** (0.12)	**2.02** (1.07)	2.01 (1.05)	1.0 (0.52)	0.57 (0.32)	1.19 (0.25)	1.17 (0.29)	**1.37** (1.17)	**1.37** (1.2)	1.02 (0.7)	0.48 (0.1)
Continuous	UAS	1.52	1.52	0.46	0.42	1.24	6.79	2.76	2.76	**0.83**	**0.88**	**5.97**	9.19
	WAS	0.38	0.38	0.17	0.15	0.7	3.18	**1.64**	1.64	1.81	1.84	8.11	**8.74**
	AutoIV	2.25 (2.68)	2.25 (2.68)	0.27 (0.24)	0.24 (0.23)	0.89 (0.68)	3.06 (1.58)	32.79 (52.83)	32.79 (52.83)	2.64 (1.85)	2.65 (1.61)	6.96 (0.44)	10.52 (2.06)
	CoCoIV(Ours)	**0.27** (0.16)	**0.29** (0.16)	**0.11** (0.09)	**0.1** (0.05)	**0.48** (0.16)	**1.87** (0.83)	2.43 (0.53)	**0.64** (0.53)	2.32 (0.65)	2.28 (0.53)	7.49 (1.13)	10.77 (2.33)

Table 11: Experiment results on assumption violation with low-dimensional synthetic datasets.

		Linear response function						Non-linear response function					
	Method	2SLS	IVGMM	DML	Ortho	Poly2SLS	KernelIV	2SLS	IVGMM	DML	Ortho	Poly2SLS	KernelIV
Binary	UAS	1.23	1.23	1.78	1.35	0.69	**0.3**	1.23	1.23	1.57	1.35	0.64	0.48
	WAS	0.91	0.91	1.4	**1.25**	0.65	0.39	1.15	1.15	1.34	1.25	0.6	0.51
	DVAE.CIV	1.5 (0.09)	1.5 (0.09)	**1.31** (1.18)	1.37 (1.18)	0.98 (0.15)	1.11 (0.05)	**1.12** (1.65)	**1.12** (1.65)	**0.6** (0.47)	1.32 (0.78)	0.88 (0.05)	0.88 (0.01)
	CoCoIV (Ours)	**0.15** (0.14)	**0.16** (0.15)	1.46 (0.48)	1.31 (0.5)	**0.59** (0.23)	0.35 (0.17)	1.46 (0.35)	1.4 (0.34)	0.8 (0.95)	**0.81** (0.9)	**0.42** (0.39)	**0.32** (0.19)
Continuous	UAS	0.47	0.47	0.12	0.14	0.43	1.7	4.39	4.39	3.79	**3.78**	**9.69**	**6.23**
	WAS	0.44	0.44	0.11	0.13	0.41	**1.11**	4.18	4.18	3.79	3.79	9.86	7.62
	AutoIV	0.51 (0.65)	0.51 (0.65)	0.14 (0.27)	0.14 (0.27)	0.66 (0.81)	2.3 (0.86)	18.41 (34.71)	18.41 (34.71)	N/A (N/A)	5.25 (5.61)	24.23 (55.97)	15.71 (14.5)
	CoCoIV (Ours)	**0.13** (0.04)	**0.13** (0.03)	**0.03** (0.01)	**0.04** (0.01)	**0.24** (0.01)	2.1 (0.33)	**3.85** (0.39)	**1.22** (0.59)	**3.77** (0.41)	3.85 (0.43)	10.66 (0.3)	19.49 (12.01)

of $U_{\mathbf{D}_1,\mathbf{D}_2}$:

$$
\begin{aligned}
&U_{\mathbf{D}_1,\mathbf{D}_2} \sim N(0,1),\\
&\mathbf{D}_1 = f_{\mathbf{D}_1}(\epsilon_{\mathbf{D}_1}) + U_{\mathbf{D}_1,\mathbf{D}_2},\\
&\mathbf{D}_2 = A^\top \mathbf{C} + \epsilon_{\mathbf{D}_2} + U \cdot 1(\mathbf{p} > 0.5) + U_{\mathbf{D}_1,\mathbf{D}_2}.
\end{aligned}
$$

Table 11[8] depicts the results on the modified datasets. Despite the violation of assumption $\mathbf{D}_1 \perp\!\!\!\perp \mathbf{D}_2$, our model records the lowest MAEs on half of the estimators with binary treatment. In addition, the decrease in performance with our model is quite mild, as shown in Table 11 compared to Table 4. All the estimators with binary treatment and linear response function even yield lower MAEs than MAEs shown in Table 4 in the paper. When MAEs are higher than in the original setting, the difference is lower than 0.1 for 2SLS, IVGMM, and Ortho with continuous treatment and linear response function. However, in the most challenging case with continuous treatment and non-linear response function, as mentioned in Sec. 3.1, all the models, including our own, yield biased estimates, leading to higher MAEs than the original setting.

Extended experiment on dependence within $\mathbf{D}_1$ and $\mathbf{D}_2$. We extend our experiments on dependence within $\mathbf{D}_1, \mathbf{D}_2$. As we do not assume independence within variables in $\mathbf{D}_1$ or within variables in $\mathbf{D}_2$, we assess how our model works in such setting.

Synthetic datasets are generated in steps similar to Sec. B.1, but we imposed dependence within $\mathbf{D}_1$ and within $\mathbf{D}_2$ as sampling

$$
\epsilon_{\mathbf{D}_1} \sim N\left(0, 10 \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}\right), \qquad \epsilon_{\mathbf{D}_2} \sim N\left(0, 0.25 \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}\right).
$$

As illustrated in Table 10, our model demonstrates better or similar MAEs for IV-based estimators 2SLS and IVGMM. In particular, our model showed relatively better performance on continuous treatment with a linear response function. Aligned with the conclusions in the previous part, performance does not decline abruptly compared to MAEs in Table 4 except for some estimators in non-linear response function.

Experiment on architecture choice As we propose a framework of generating representation of IVs, one can select a range of architectures with the same framework. For reference, we provide results of different combinations of latent variable models: 1) VAEs for encoders of $\mathbf{Z}, \mathbf{C}$, 2) VaDEs for encoders of $\mathbf{Z}, \mathbf{C}$. Table 12 demonstrates that, regardless of the choice of architecture, the models derive a similar range of estimates for each estimator. In real-world applications, the choice of architecture can be varied according to the property of underlying data.

[8]Here the result of DML with AutoIV on continuous treatment and non-linear response function shows significantly high MAE over 100,000, likely due to the sensitivity of DML mentioned in Sec. 3.1

Table 12: Experiment results on various latent variable models.

		Linear response function						Non-linear response function					
	Method	2SLS	IVGMM	DML	Ortho	Poly2SLS	KernelIV	2SLS	IVGMM	DML	Ortho	Poly2SLS	KernelIV
Binary	Ours	**0.26** (0.09)	**0.31** (0.07)	**1.59** (1.17)	**1.3** (1.25)	1.66 (3.24)	0.38 (0.31)	**1.12** (0.17)	**1.06** (0.2)	**0.4** (0.56)	**0.37** (0.47)	**0.47** (0.18)	**0.33** (0.02)
	only VAE	0.43 (0.36)	0.45 (0.36)	1.96 (1.68)	1.75 (1.48)	**0.99** (0.58)	0.51 (0.36)	1.23 (0.84)	1.14 (0.85)	1.15 (0.95)	1.28 (1.54)	0.75 (0.36)	0.4 (0.12)
	only VaDE	0.28 (0.06)	**0.31** (0.05)	2.32 (2.0)	2.0 (1.37)	1.06 (0.86)	**0.34** (0.2)	1.28 (0.11)	1.26 (0.09)	1.26 (1.08)	1.06 (0.88)	0.6 (0.43)	0.38 (0.14)
Continuous	Ours	**0.07** (0.09)	**0.08** (0.09)	**0.03** (0.03)	**0.02** (0.01)	**0.35** (0.07)	**3.51** (0.55)	1.38 (0.1)	0.3 (0.15)	1.35 (0.12)	1.37 (0.1)	**4.13** (0.32)	3.82 (1.16)
	only VAE	0.09 (0.1)	0.11 (0.12)	**0.03** (0.04)	0.03 (0.02)	0.37 (0.11)	3.61 (0.38)	**1.33** (0.1)	0.35 (0.3)	**1.32** (0.07)	**1.35** (0.07)	4.24 (0.4)	**3.59** (0.87)
	only VaDE	0.16 (0.17)	0.17 (0.19)	0.05 (0.04)	0.06 (0.05)	0.44 (0.27)	3.61 (0.54)	1.47 (0.13)	**0.26** (0.16)	1.43 (0.11)	1.44 (0.08)	4.3 (0.47)	4.09 (1.56)

In our case, we adopt a VAE-based framework with flexible prior [Sohn et al., 2015] to encode $\mathbf{Z}$ that are associated with $\mathbf{D}_1$. For the prior, we use $p(\mathbf{z}) \approx \frac{1}{N}\sum_{i=1}^{N} p(\mathbf{z} \mid \mathbf{d}^{(i)})$ where $\mathbf{d}^{(i)} \sim p(\mathbf{d})$. We chose $N = 1$ and intended to learn the prior $p(\mathbf{z}) \approx p(\mathbf{z} \mid \mathbf{d})$. Also, we aim to encode representation $\mathbf{C}$ that can capture likely a complex structure of non-IV $\mathbf{D}_2$, which usually have a number more than that of true IVs in observed covariates in real-world scenarios. Thus, we model prior $p(\mathbf{c})$ as a Gaussian Mixture Model, adopting Jiang et al. [2017]. By accommodating the mixture model prior, we aim to learn $\mathbf{C}$ expressing complex latent structures of the observed data.