Correcting the Unobserved Textual Confounding Factors in Regression by BERT and Partially Linear Regression–Online Appendix

Mengke Qiao
Culverhouse College of Business, University of Alabama, Tuscaloosa, AL, U.S.A., mqiao@ua.edu
Ke-Wei Huang
School of Computing, National University of Singapore, Singapore, dishkw@nus.edu.sg

Appendix A: Proof for DML, C-DML, and GRACE

Before we introduce the proof in detail, we first define the model specification. The semi-parametric linear regression is defined by the following two equations,

$$Y = \theta D + g(X) + U, \text{ where } \mathrm{E}(U|D, X) = 0; \tag{28}$$

$$D = m(X) + V, \text{ where } \mathrm{E}(V|X) = 0. \tag{29}$$

Here, Y is the dependent variable (DV), D is the independent variable (IV) of interest, X is a vector of the potential control variables (e.g., text embedding and other numerical variables), and $g(X)$ and $m(X)$ are unknown functions. Our goal is to estimate the parameter of interest θ.

Next, we take expectations conditional on X for Eq.(28),

$$\mathrm{E}(Y|X) = \theta \mathrm{E}(D|X) + g(X). \tag{30}$$

We subtract Eq.(30) from Eq.(28) to obtain,

$$Y - \mathrm{E}(Y|X) = \theta(D - \mathrm{E}(D|X)) + U. \tag{31}$$

To simplify notations, $\mathrm{E}(Y|X)$ is denoted as $l(X)$, and $\mathrm{E}(D|X)$ is denoted as $m(X)$. Given Eq.(31), DML proceeds with the following estimation steps.

1. $m(X)$ and $l(X)$ are estimated by using control variables to predict independent and dependent variables.
2. Given the estimated $\hat{m}(X)$ and $\hat{l}(X)$, two residuals are derived, $\hat{V} = D - \hat{m}(X)$ and $Y - \hat{l}(X)$.

3. Using $\hat{V}$ as IV and $Y-\hat{l}(X)$ as DV to derive θ by OLS estimator,

$$\bar{\theta}=(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2)^{-1}\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i(Y_i-\hat{l}(X_i)). \tag{32}$$

Bias Analysis for DML

Let $\Delta m(X_i)=m(X_i)-\hat{m}(X_i)$, $\Delta l(X_i)=l(X_i)-\hat{l}(X_i)$, and $\hat{V}_i=D_i-\hat{m}(X_i)=V_i+\Delta m(X_i)$.

DML estimator yields,

$$\begin{aligned}
\bar{\theta}&=(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2)^{-1}\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i(Y_i-\hat{l}(X_i))\\
&=(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2)^{-1}\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i(\hat{V}_i\theta-\Delta m(X_i)\theta+U_i+\Delta l(X_i))\\
&=(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2)^{-1}\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2\theta-V_i\Delta m(X_i)\theta-\Delta m^2(X_i)\theta+U_iV_i+\Delta m(X_i)U_i+V_i\Delta l(X_i)+\Delta m(X_i)\Delta l(X_i)\\
&=\theta+(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2)^{-1}\frac{1}{n}\sum_{i=1}^{n}\underbrace{\Delta m(X_i)\Delta l(X_i)}_{\text{Regularization bias}}-\underbrace{V_i\Delta m(X_i)\theta+\Delta m(X_i)U_i+V_i\Delta l(X_i)}_{\text{Overfitting bias}}-\underbrace{\Delta m^2(X_i)\theta}_{\text{Remaining term}}.
\end{aligned} \tag{33}$$

Since V_i and U_i are uncorrelated, $\frac{1}{n}\sum_{i=1}^{n}V_iU_i=0$. Regularization bias is explained in Section 2.4. The second important bias term is induced by the overfitting nature of machine learning algorithms. Specifically, $\Delta m(X_i)$ (and $\Delta l(X_i)$) may be correlated with U_i (and V_i) and lead to the overfitting bias,

$$c_{overfit}=(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2)^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}-\theta V_i\Delta m(X_i)+\Delta m(X_i)U_i+V_i\Delta l(X_i). \tag{34}$$

To eliminate overfitting bias, Chernozhukov et al. (2018) proposed randomly splitting the sample into two sub-samples, I_1 and I_2. The sub-sample I_1 is utilized to estimate $m(X)$ and $l(X)$ while θ is estimated by utilizing I_2. In this case, this bias can be reduced since the overfitting error in $\hat{m}(X)$ and $\hat{l}(X)$ is obtained from I_1 while U_i and V_i are from an independent sub-sample I_2. Given this sample splitting condition and the convergence rate requirements, Chernozhukov et al. (2018) proved that DML can achieve consistency for coefficient estimation.

Bias Analysis for C-DML

The loss function for C-DML includes three terms, prediction loss terms for $m(X)$ and $l(X)$, and one regularization term $\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i(Y-\hat{l}(X))$. Asymptotically, this regularization term equals, $\frac{1}{n}\sum_{i=1}^{n}\Delta m(X_i)\Delta l(X_i)+\theta V_i^2$, which is shown in Eq.(35).

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i(Y_i-\hat{l}(X_i))&=\frac{1}{n}\sum_{i=1}^{n}(V_i+\Delta m(X_i))(\Delta l(X_i)+\theta V_i+U_i)\\
&=\frac{1}{n}\sum_{i=1}^{n}V_iU_i+\Delta m(X_i)U_i+V_i\Delta l(X_i)+\Delta m(X_i)\Delta l(X_i)+\theta V_i^2+\theta\Delta m(X_i)V_i\\
&=\frac{1}{n}\sum_{i=1}^{n}\Delta m(X_i)\Delta l(X_i)+\theta V_i^2.
\end{aligned} \tag{35}$$

Since C-DML utilizes data splitting of machine learning and regression datasets, $\frac{1}{n}\sum_{i=1}^{n}\Delta m(X_i)U_i+V_i\Delta l(X_i)+\theta\Delta m(X_i)V_i$ can be eliminated. Moreover, since V_i and U_i are uncorrelated, $\frac{1}{n}\sum_{i=1}^{n}V_iU_i=0$. Therefore, $\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i(Y_i-\hat{l}(X_i))=\frac{1}{n}\sum_{i=1}^{n}\Delta m(X_i)\Delta l(X_i)+\theta V_i^2$.

The main goal of this term is to reduce the regularization bias in Eq.(8). However, only when $\theta = 0$, this regularization term equals the regularization bias. If θ is positive, $\frac{1}{n}\sum_{i=1}^{n}\Delta m(X_i)\Delta l(X_i)$ may be optimized to a negative value and the coefficient will be underestimated. If θ is negative, $\frac{1}{n}\sum_{i=1}^{n}\Delta m(X_i)\Delta l(X_i)$ may be optimized to a positive value and the coefficient will be overestimated.

Bias Analysis for GRACE method The loss function for GRACE method includes three terms, prediction loss terms for $m(X)$ and $l(X)$, and one innovative loss term as follows.

$$
\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i(Y_i-(D-\hat{m}(X_i))\hat{\theta}-\hat{l}(X_i))\\
&=\frac{1}{n}\sum_{i=1}^{n}(V_i+\Delta m(X_i))(V_i\theta+l(X_i)+U_i-\hat{V}_i\hat{\theta}-\hat{l}(X_i))\\
&=\frac{1}{n}\sum_{i=1}^{n}(V_i+\Delta m(X_i))(V_i\theta+\Delta l(X_i)+U_i-\hat{V}_i\hat{\theta})\\
&=\frac{1}{n}\sum_{i=1}^{n}(V_i^2\theta+\Delta m(X_i)V_i\theta+\Delta l(X_i)V_i+\Delta m(X_i)\Delta l(X_i)+V_iU_i+U_i\Delta m(X_i)-\hat{V}_i\hat{\theta}V_i-\hat{V}_i\hat{\theta}\Delta m(X_i))\\
&=\frac{1}{n}\sum_{i=1}^{n}(V_i^2\theta+\Delta m(X_i)V_i\theta+\Delta l(X_i)V_i+\Delta m(X_i)\Delta l(X_i)+V_iU_i+U_i\Delta m(X_i)-(V_i+\Delta m(X_i))\hat{\theta}V_i\\
&-(V_i+\Delta m(X_i))\hat{\theta}\Delta m(X_i))\\
&=\frac{1}{n}\sum_{i=1}^{n}(V_i^2(\theta-\hat{\theta})+\Delta m(X_i)V_i\theta+\Delta l(X_i)V_i+\Delta m(X_i)\Delta l(X_i)+V_iU_i+U_i\Delta m(X_i)-2\Delta m(X_i)V_i\hat{\theta}-\Delta m^2(X_i)\hat{\theta})\\
&=\frac{1}{n}\sum_{i=1}^{n}(V_i^2(\theta-\hat{\theta})+\Delta m(X_i)\Delta l(X_i)-\Delta m^2(X_i)\hat{\theta}+\Delta m(X_i)V_i\theta+\Delta l(X_i)V_i+V_iU_i+U_i\Delta m(X_i)-2\Delta m(X_i)V_i\hat{\theta})\\
&=\frac{1}{n}\sum_{i=1}^{n}(V_i^2(\theta-\hat{\theta})+\Delta m(X_i)\Delta l(X_i)-\Delta m^2(X_i)\hat{\theta}).
\end{aligned}
\tag{36}
$$

Since our method has utilized data splitting of the training set and validation set to avoid overfitting, $\frac{1}{n}\sum_{i=1}^{n}(\Delta m(X_i)V_i\theta+\Delta l(X_i)V_i+U_i\Delta m(X_i)-2\Delta m(X_i)V_i\hat{\theta})$ can also be eliminated. Moreover, since V_i and U_i are uncorrelated, $\frac{1}{n}\sum_{i=1}^{n}V_iU_i=0$. Therefore, the final loss equals $\frac{1}{n}\sum_{i=1}^{n}(V_i^2(\theta-\hat{\theta})+\Delta m(X_i)\Delta l(X_i)-\Delta m^2(X_i)\hat{\theta})$. Comparing our loss with C-DML loss, we can find that we minimize the difference between θ and $\hat{\theta}$. Moreover, we also minimize the regularization bias term and the remaining term of DML in Eq.(36) by $\frac{1}{n}\sum_{i=1}^{n}\Delta m(X_i)\Delta l(X_i)-\Delta m^2(X_i)\hat{\theta}$.

Gradient Derivation for DML, C-DML, and Our Method

For semi-parametric linear regression, DML separately estimates $m(X)$ and $l(X)$. To simplify the derivation process, we assume both prediction problems utilize MSE loss.

DML: Separate Optimization

$$
l_m=\frac{1}{n}\sum_{i=1}^{n}(D_i-\hat{m}(X_i))^2,\frac{\partial l_m}{\partial\alpha_m}=-2\frac{1}{n}\sum_{i=1}^{n}(D_i-\hat{m}(X_i))\frac{\partial\hat{m}(X_i)}{\partial\alpha_m},
\tag{37}
$$

$$
l_l=\frac{1}{n}\sum_{i=1}^{n}(Y_i-\hat{l}(X_i))^2,\frac{\partial l_l}{\partial\alpha_l}=-2\frac{1}{n}\sum_{i=1}^{n}(Y_i-\hat{l}(X_i))\frac{\partial\hat{l}(X_i)}{\partial\alpha_l},
\tag{38}
$$

where l_m and l_l denote the MSE loss for $\hat{m}(X)$ and $\hat{l}(X)$, α_m and α_l denote the parameters for $\hat{m}(X)$ and $\hat{l}(X)$.

C-DML: Joint Estimation of $m(X)$ and $l(X)$

$$L = \frac{1}{n}\sum_{i=1}^{n}(D_i - \hat{m}(X_i))^2 + \frac{1}{n}\sum_{i=1}^{n}[Y_i - \hat{l}(X_i)]^2 + [\frac{1}{n}\sum_{i=1}^{n}[Y_i - \hat{l}(X_i)][D_i - \hat{m}(X_i)]]^2. \tag{39}$$

$$\begin{aligned}\frac{\partial L}{\partial \alpha_m} = &-2\frac{1}{n}\sum_{i=1}^{n}(D_i - \hat{m}(X_i))\frac{\partial \hat{m}(X_i)}{\partial \alpha_m} \\ &+ 2(\frac{1}{n}\sum_{i=1}^{n}[Y_i - \hat{l}(X_i)][D_i - \hat{m}(X_i)])(\frac{1}{n}\sum_{i=1}^{n} -[Y_i - \hat{l}(X_i)]\frac{\partial \hat{m}(X_i)}{\partial \alpha_m}).\end{aligned} \tag{40}$$

$$\begin{aligned}\frac{\partial L}{\partial \alpha_l} = &-2\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{l}(X_i))\frac{\partial \hat{l}(X_i)}{\partial \alpha_l} \\ &+ 2(\frac{1}{n}\sum_{i=1}^{n}[Y_i - \hat{l}(X_i)][D_i - \hat{m}(X_i)])(\frac{1}{n}\sum_{i=1}^{n} -[D_i - \hat{m}(X_i)]\frac{\partial \hat{l}(X_i)}{\partial \alpha_l}).\end{aligned} \tag{41}$$

GRACE Method: Joint Estimation of $m(X)$, $l(X)$, and θ

$$\begin{aligned}L = \frac{1}{n}\sum_{i=1}^{n}(D_i - \hat{m}(X_i))^2 + \frac{1}{n}\sum_{i=1}^{n}[Y_i - \hat{\theta}(D_i - \hat{m}(X_i)) - \hat{l}(X_i)]^2 \\ + [\frac{1}{n}\sum_{i=1}^{n}[Y_i - \hat{\theta}(D_i - \hat{m}(X_i)) - \hat{l}(X_i)][D_i - \hat{m}(X_i)]]^2.\end{aligned} \tag{42}$$

Let $\eta_i = Y_i - \hat{\theta}(D_i - \hat{m}(X_i)) - \hat{l}(X_i)$, then:

$$\begin{aligned}\frac{\partial L}{\partial \alpha_m} = &-2\frac{1}{n}\sum_{i=1}^{n}(D_i - \hat{m}(X_i))\frac{\partial \hat{m}(X_i)}{\partial \alpha_m} + 2\frac{1}{n}\sum_{i=1}^{n}\eta_i\hat{\theta}\frac{\partial \hat{m}(X_i)}{\partial \alpha_m} \\ &+ 2(\frac{1}{n}\sum_{i=1}^{n}\eta_i[D_i - \hat{m}(X_i)])(\frac{1}{n}\sum_{i=1}^{n}((D_i - \hat{m}(X_i))\hat{\theta}\frac{\partial \hat{m}(X_i)}{\partial \alpha_m}) \\ &+ 2(\frac{1}{n}\sum_{i=1}^{n}\eta_i[D_i - \hat{m}(X_i)])(\frac{1}{n}\sum_{i=1}^{n} -\eta_i\frac{\partial \hat{m}(X_i)}{\partial \alpha_m}).\end{aligned} \tag{43}$$

$$\frac{\partial L}{\partial \alpha_l} = -2\frac{1}{n}\sum_{i=1}^{n}\eta_i\frac{\partial \hat{l}(X_i)}{\partial \alpha_l} - 2(\frac{1}{n}\sum_{i=1}^{n}\eta_i[D_i - \hat{m}(X_i)])(\frac{1}{n}\sum_{i=1}^{n}((D_i - \hat{m}(X_i))\frac{\partial \hat{l}(X_i)}{\partial \alpha_l}). \tag{44}$$

$$\frac{\partial L}{\partial \theta} = -2\frac{1}{n}\sum_{i=1}^{n}\eta_i(D_i - \hat{m}(X_i)) - 2(\frac{1}{n}\sum_{i=1}^{n}\eta_i[D_i - \hat{m}(X_i)])(\frac{1}{n}\sum_{i=1}^{n}(D_i - \hat{m}(X_i))^2). \tag{45}$$

Comparing the gradients from the three methods, we can find that gradients from our method indeed utilize the full information and consider the interdependence of $m(X)$, $l(X)$, and θ in the whole system. Therefore, it is very likely that our method can estimate $m(X)$, $l(X)$, and θ more accurately.

Appendix B: Additional Results for Semi-parametric Linear Regression

In this section, we report the additional results for semi-parametric linear regression by using the Bert-large-uncased model, the Distilbert-base-uncased model, and the FinBERT model to generate text embedding. The results are shown in Table 9. We also report the estimated coefficient distribution in Figure 7. The conclusions are qualitatively similar to Section 4.

1. **Bert-large-uncased model.** This model stacks 24 layers (i.e., transformer blocks) with 16 bidirectional self-attention heads. It has 340M parameters with a hidden size of 1024. This model is uncased: it does not make a difference between lowercase and uppercase letters.

Table 1 Additional Results for Semi-parametric Linear Regression

Case	$f()$	β	Metric	Naïve	Manual	Bert-large		Distilbert		FinBERT	
						GRACE	DML	GRACE	DML	GRACE	DML
1	Linear	5	Bias	0.209	0.182	0.035	0.091	0.009	0.056	0.021	0.076
			Std	0.014	0.014	0.027	0.022	0.023	0.022	0.019	0.019
			MSE	0.044	0.033	0.002	0.008	0.000	0.003	0.001	0.006
2	Square	5	Bias	0.207	0.176	0.030	0.096	0.016	0.073	0.010	0.078
			Std	0.014	0.014	0.031	0.023	0.024	0.022	0.025	0.020
			MSE	0.043	0.031	0.001	0.009	0.000	0.005	0.000	0.006
3	Linear	10	Bias	0.422	0.362	0.029	0.158	0.027	0.067	0.015	0.117
			Std	0.019	0.020	0.039	0.029	0.034	0.032	0.038	0.028
			MSE	0.178	0.131	0.002	0.025	0.001	0.005	0.001	0.014
4	Square	10	Bias	0.424	0.345	0.010	0.167	0.032	0.082	0.017	0.130
			Std	0.020	0.020	0.053	0.035	0.037	0.032	0.033	0.029
			MSE	0.179	0.119	0.002	0.028	0.002	0.007	0.001	0.017

2. **Distilbert-base-uncased model.** This model is a distilled version of the BERT base model. It reduces the size of a BERT model by 40%, while retaining 97% of its language understanding capabilities and being 60% faster.

3. **FinBERT.** FinBERT is a BERT model pre-trained on financial communication text.

We also report the experiment details and comparison results for our method, DML, C-DML, Veitch et.al (2020), and Shi et al.(2019) by using BERT Embedding in Table 10.

1. **Coordinated Double Machine Learning (C-DML)**. For this method, we follow the experiment settings of DML. The difference is we train the network with the C-DML loss function.

2. **Veitch et al. (2020)**. For this method, we train the neural network model by using two prediction loss terms for the outcome and treatment jointly. The model has three outputs: propensity score, and outcomes under both treatment and control conditions. Other settings are similar to DML. [1]

3. **Shi et al. (2019)**. The loss function for this method includes three terms: two prediction loss terms for the outcome and treatment, and one regularization loss based on non-parametric theory. Following Shi et al. (2019), we exclude data points with estimated propensity scores outside [0.01, 0.99] and use all the data to calculate the treatment effect. Other settings are similar to Veitch et al. (2020).

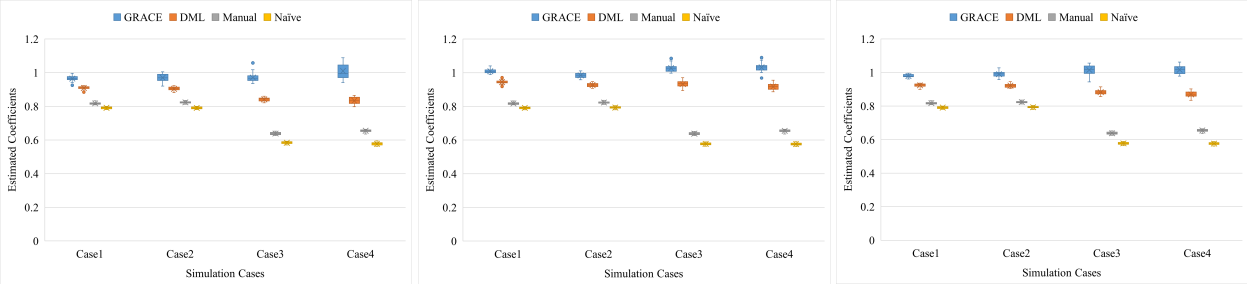
We have utilized both Adam and SGD as optimizers. The results consistently show that our method has overall better performance than all the benchmark methods. Our method outperforms C-DML and Veitch et al. (2020) because we have integrated prediction and estimation together with one novel loss based on econometric theory to reduce the estimation bias. Our method outperforms Shi et al. (2019) maybe because this method is sensitive to the propensity score, which is in the denominator of the estimator.

Appendix C: Additional Results for Semi-parametric Logistic Regression

In this section, we report the additional results for semi-parametric logistic regression by using the Bert-large-uncased model, the Distilbert-base-uncased model, and the FinBERT model to generate text embedding.

[1] Slightly different from Veitch et al. (2020), we do not include the masked language modeling objective and allow updates of BERT embedding since all the benchmark methods use the fixed text representations.

Figure 1 Additional Results Comparison for Linear Regression



(a) Bert-large-uncased Results (b) Distilbert Results (c) Finbert Results

*Note.*The legend entries are the same as Figure 3.

Table 2 Additional Neural Network-Based Methods Comparison Results

Optimizer	Case	GRACE		DML		C-DML		Veitch et.al(2020)		Shi et.al(2019)	
		Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE
Adam	Case1	0.056	0.004	0.163	0.027	0.073	0.006	0.009	0.006	0.041	0.008
	Case2	0.012	0.001	0.178	0.032	0.107	0.012	0.048	0.008	0.089	0.021
	Case3	0.049	0.014	0.301	0.093	0.160	0.027	0.132	0.027	0.158	0.046
	Case4	0.066	0.006	0.319	0.102	0.150	0.025	0.163	0.035	0.157	0.039
SGD	Case1	0.023	0.001	0.077	0.006	0.080	0.007	0.046	0.004	0.004	0.070
	Case2	0.051	0.003	0.105	0.012	0.110	0.013	0.078	0.010	0.075	0.063
	Case3	0.071	0.006	0.139	0.020	0.155	0.025	0.179	0.040	0.157	0.140
	Case4	0.053	0.004	0.135	0.019	0.149	0.024	0.164	0.029	0.154	0.127

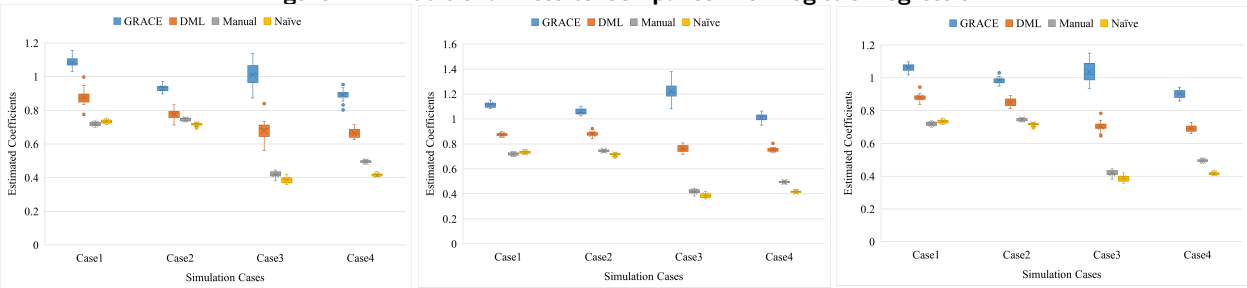
Table 3 Additional Results for Semi-parametric Logistic Regression

Case	$f()$	β	Metric	Naïve	Manual	Bert-large		Distilbert		FinBERT	
						GRACE	DML	GRACE	DML	GRACE	DML
1	Linear	5	Bias	0.265	0.281	0.085	0.122	0.113	0.126	0.063	0.119
			Std	0.023	0.024	0.049	0.062	0.045	0.037	0.039	0.038
			MSE	0.071	0.079	0.008	0.017	0.013	0.016	0.004	0.015
2	Square	5	Bias	0.283	0.253	0.068	0.223	0.059	0.119	0.015	0.149
			Std	0.024	0.024	0.049	0.054	0.048	0.044	0.040	0.042
			MSE	0.080	0.064	0.005	0.051	0.004	0.015	0.001	0.023
3	Linear	10	Bias	0.613	0.580	0.012	0.320	0.222	0.239	0.038	0.297
			Std	0.040	0.038	0.102	0.087	0.100	0.061	0.088	0.061
			MSE	0.376	0.337	0.005	0.106	0.053	0.058	0.005	0.089
4	Square	10	Bias	0.582	0.504	0.108	0.336	0.014	0.246	0.097	0.308
			Std	0.022	0.023	0.054	0.047	0.050	0.039	0.043	0.037
			MSE	0.339	0.254	0.013	0.114	0.001	0.061	0.010	0.095

The results are shown in Table 11. We also report the estimated coefficient distribution in Figure 8. The conclusions are qualitatively similar to Section 4.

We also report the estimation results for low confounding level when $\beta = 1$ in Section 4.4 and Section 4.5. The results are shown in Table 12. Generally, the results show that when the confounding level is low, all the methods have similar performance. Our method will not hurt the estimation of beta.

Figure 2 Additional Results Comparison for Logistic Regression



(a) Bert-large-uncased Results (b) Distilbert Results (c) Finbert Results

*Note.*The legend entries are the same as Figure 3.

Table 4 Additional Results for Low Confounding Level

Model	$f()$	β	Metric	Naïve	Manual	Embedding		TF-IDF		Topic	
						GRACE	DML	GRACE	DML	GRACE	DML
Linear	Linear	1	Bias	0.040	0.038	0.006	0.033	0.004	0.021	0.024	0.025
			Std	0.011	0.005	0.022	0.020	0.018	0.016	0.014	0.013
			MSE	0.002	0.001	0.000	0.001	0.000	0.001	0.001	0.001
	Square	1	Bias	0.038	0.039	0.015	0.039	0.008	0.023	0.026	0.025
			Std	0.011	0.005	0.019	0.018	0.016	0.016	0.015	0.013
			MSE	0.002	0.002	0.000	0.002	0.000	0.001	0.001	0.001
Logit	Linear	1	Bias	0.033	0.007	0.049	0.034	0.053	0.063	0.030	0.037
			Std	0.035	0.014	0.069	0.080	0.065	0.046	0.042	0.038
			MSE	0.001	0.000	0.003	0.003	0.004	0.004	0.001	0.002
	Square	1	Bias	0.078	0.094	0.066	0.069	0.069	0.035	0.025	0.042
			Std	0.038	0.017	0.085	0.074	0.070	0.053	0.046	0.044
			MSE	0.006	0.009	0.005	0.005	0.006	0.002	0.001	0.002

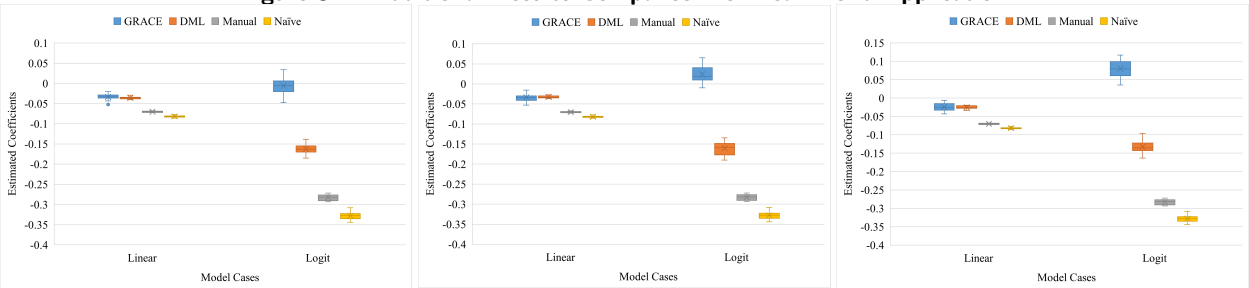
Table 5 Additional Results for Effect of "the" on Review Subjectivity

Model	Metric	Naïve	Manual	BERT-large		Distilbert		FinBERT	
				GRACE	DML	GRACE	DML	GRACE	DML
Linear	Bias	0.082	0.070	0.034	0.036	0.033	0.032	0.025	0.025
	Std	0.005	0.005	0.009	0.006	0.011	0.006	0.012	0.008
	MSE	0.007	0.005	0.001	0.001	0.001	0.001	0.001	0.001
Logit	Bias	0.329	0.282	0.006	0.162	0.024	0.161	0.080	0.132
	Std	0.020	0.020	0.034	0.027	0.037	0.032	0.044	0.036
	MSE	0.108	0.079	0.000	0.026	0.001	0.026	0.007	0.018

Appendix D: Additional Results for Real-World Application

In this section, we report the additional results for the real-world application by using the Bert-large-uncased model, the Distilbert-base-uncased model, and the FinBERT model to generate text embedding. The results are shown in Table 13. We also report the estimated coefficient distribution in Figure 9. The conclusions are qualitatively similar to Section 5.

Figure 3 Additional Results Comparison for Real-World Application



(a) Bert-large-uncased Results (b) Distilbert Results (c) Finbert Results

*Note.*The legend entries are the same as Figure 3.

Table 6 Estimation Results for Cash Bonus Application

	Naïve	DML	GRACE
Coefficient	-0.086	-0.076	-0.079
Std	0.036	0.044	0.051

Appendix E: Real-World Application 2

We have conducted another real-world application using the published dataset in the Double Machine Learning paper (Chernozhukov et al. 2018). Since the dataset is from a random assigned field experiment, control variables may not affect the coefficient, so this setting may not have omitted variable bias. We want to use this experiment to show the robustness of our method across different settings.

In their empirical application, Chernozhukov et al. (2018) examined the incentive impact of compensation schemes for unemployment insurance (UI). Specifically, in their setting, the control group followed the regular UI rules, while the treatment group was offered a generous cash bonus for finding and keeping a job within a long qualification period, and optional workshop participation. For this analysis, the key variables included a treatment assignment indicator as the key explanatory variable and the outcome variable, which is the logarithm of the unemployment duration for UI claimants. The covariates in this study encompassed a range of demographic and employment-related factors, including age, gender, race, number of dependents, timing of the experiment within the year, geographic location in the state, anticipated job recall, and occupational type.

Since the data comes from a random experiment, theoretically, the beta coefficient from DML and our method should be similar to that of the naïve method without any correction. The results are described in Table 14. The results show that beta coefficients from our method and DML are indeed very similar and slightly smaller in terms of size compared to the beta from the Naïve method. These results demonstrate the robustness of our method in this kind of real-world application.

Appendix F: Additional Generalizability Results in Different Domains

To enhance the generalizability of our method in different domains and different unstructured data types, we have also conducted the following two experiments.

In the first experiment, we utilize Twitter text data to evaluate our method. The simulation process is as follows. First, we generate text embedding vectors from 50000 tweets using the BERT model. The final

Table 7 Estimation Results for Twitter Data

β	$f()$	Linear Model				Logit Model			
		Naïve		GRACE		Naïve		GRACE	
		Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE
1	Linear	0.090	0.008	0.011	0.000	0.139	0.020	0.034	0.002
1	Square	0.094	0.009	0.008	0.000	0.231	0.054	0.012	0.001
5	Linear	0.466	0.217	0.008	0.000	0.726	0.526	0.035	0.004
5	Square	0.500	0.250	0.043	0.002	0.850	0.723	0.049	0.004
10	Linear	0.914	0.836	0.065	0.006	0.912	0.831	0.044	0.006
10	Square	0.967	0.936	0.056	0.003	1.059	1.121	0.016	0.009

Table 8 Estimation Results for Airbnb Data

β	$f()$	Linear Model				Logit Model			
		Naïve		GRACE		Naïve		GRACE	
		Bias	MSE	Bias	MSE	Bias	MSE	Bias	MSE
1	Linear	0.027	0.001	0.015	0.000	0.088	0.008	0.035	0.002
1	Square	0.018	0.000	0.011	0.000	0.072	0.006	0.059	0.005
5	Linear	0.167	0.028	0.023	0.001	0.256	0.066	0.025	0.002
5	Square	0.095	0.009	0.013	0.000	0.097	0.010	0.062	0.007
10	Linear	0.335	0.112	0.089	0.010	0.499	0.249	0.029	0.003
10	Square	0.180	0.032	0.089	0.008	0.324	0.105	0.016	0.001

embedding vectors have 768 dimensions. Next, we randomly select two dimensions, $C_1$and C_2, as the omitted control variables. We construct a new variable $Z = C_1 + C_2$. Then we generate the focal independent variable D following the equation, $P(D = 1|Z) = \frac{exp(Z)}{1+exp(Z)}$. Finally, for linear model, we generate the dependent variable Y by $Y = 2 + 1 \times D + g(Z) + \epsilon$, where ϵ follows the standard normal distribution. For logit model, we generate the dependent variable Y by $LogitP(Y = 1) = 2 + 1 \times D + g(Z)$. $g(Z) = -\beta \times f(Z)$ where β is set as 1, 5, or 10, and $f()$ is set as a linear or quadratic function. The estimation bias and MSE results are shown in Table 15. In this table, we mainly compare the naïve method and our method to evaluate whether our method can correct the estimation bias. When we apply our method, we use the text embedding from the BERT model as input in our method to capture the potential control variables. The results show that our method indeed can correct the estimation bias in the Twitter domain.

In the second experiment, we utilize Airbnb Image data to evaluate our method. The simulation process is as follows. First, we generate image embedding vectors from 50000 Airbnb images using the EfficientNet model. The final embedding vectors have 1280 dimensions. Next, we randomly select ten dimensions, C_1, C_2...,C_{10}, as the omitted control variables. We construct a new variable $Z = 0.2 \times \sum_{i=1}^{10} C_i$. Then we generate the focal independent variable D following the equation, $P(D = 1|Z) = \frac{exp(Z)}{1+exp(Z)}$. Finally, we generate the dependent variable Y using the same process as in the first experiment. When we apply our method, we use the image embedding from the EfficientNet model as input in our method to capture the potential control variables. The results are shown in Table 16. The results show that our method indeed can correct the estimation bias when embedding vectors are generated based on Airbnb images.

Appendix G: Practical Implementation for Our Method

Input. Text embeddings generated from a pre-trained BERT model, along with other observed control variables, the focal independent variable (IV), and the dependent variable (DV).

Step 1: Assess predictive strength and perform dimension reduction if needed

Step 1.1: Estimate predictive strength.

Predict the focal independent variable using the available controls and embeddings and compute R^2.

Step 1.2: Evaluate the level of redundancy.

If R^2 is below 0.8, proceed directly with the GRACE or the DML approach using the original set of controls and embeddings, and continue to Step 2.

Step 1.3: Apply dimension reduction (if necessary).

If R^2 is higher than 0.8, we recommend that researchers apply the proposed dimension reduction procedure to eliminate redundant features and utilize the newly constructed features as input of our GRACE method or the DML method.

Step 2: Specify the outcome model.

Researchers then select the appropriate model specification based on the nature of the dependent variable. For a linear outcome model, the estimation follows the procedure illustrated in Section 3.1. For a binary outcome model, the estimation should follows the procedure in Section 3.2.

Output. The procedure yields an estimate of the parameter of interest, along with its associated variance.

Appendix H: Inference Procedure Evaluation

GRACE departs from classical cross-fitting by using joint estimation with train/validation splits and early stopping. We have conducted two simulation experiments to check the performance of the current inference approach in our paper.

The first simulation process is as follows. Using a sample size of 50000, we first simulate 20 variables following a standard normal distribution to construct the potential control variable list $C = (C_1, C_2, \ldots, C_{20})$. Next, we simulate the focal independent variable

$$D = C_1 + C_2 + \varepsilon,$$

where ε follows a standard normal distribution. Finally, we simulate the dependent variable

$$Y = 2 + D - C_1 - C_2 + \varepsilon_2,$$

where ε_2 also follows a standard normal distribution. Variables C_1 and C_2 serve as omitted control variables, while the potential control variable list $C = (C_1, C_2, \ldots, C_{20})$ is used to capture them in our method. We employ two methods to estimate the standard error of the coefficient.

a. Theoretical Method: We repeat the above simulation procedure 20 times and obtain 20 datasets. For each dataset, we randomly split it into a training set and a validation set and apply our method. As a result, we obtain 20 beta coefficients. Next, we calculate the standard error of 20 beta coefficients. This standard error is the theoretical standard error of the coefficient due to the variation of the data sampling process.

Table 9 Standard Error Results Comparison

	Theoretical Method		Empirical Method	
	Coefficient	Standard Error	Coefficient	Standard Error
Case 1	0.999	0.005	0.999	0.005
Case 2	1.002	0.014	1.006	0.014

b. Empirical Method: We only conduct the above simulation procedure one time and obtain one dataset. For this dataset, we repeat the splitting of the training set and validation set 20 times, yielding 20 beta coefficients. Next, we apply the theoretical formula in Eq.(15) in our paper to calculate the standard error. This standard error is the empirical standard error from the formula in our method. The results are reported in the "Case 1" row of Table 17. The results show that our method has a similar standard error to the theoretical standard error.

The second simulation process is the same as the first simulation process except that D is simulated as a binary variable. The simulation equation is

$$P(D=1 \mid C_1, C_2) = \frac{\exp(C_1 + C_2 + \varepsilon_3)}{1 + \exp(C_1 + C_2 + \varepsilon_3)},$$

where ε_3 follows a standard normal distribution. We still use the two methods above to estimate the standard error of the coefficient. The results are reported in the "Case 2" row of Table 17. The results show that our method still produces a standard error similar to the theoretical standard error in this setting.

References

Shi C, Blei D, Veitch V (2019) Adapting neural networks for the estimation of treatment effects. *Advances in Neural Information Processing Systems* 32:2507–2517.

Veitch V, Sridhar D, Blei D (2020) Adapting text embeddings for causal inference. *Conference on Uncertainty in Artificial Intelligence*, 919–928.