Supplement Material for 'DebiNet: Debiasing Linear Models with Nonlinear Overparameterized Neural Networks'

A Preliminaries of Partially Linear Models

In this section, we revisit some partially linear models from an algorithmic perspective. The first one is the PLM with kernel regressions.

Algorithm 3 Partially Linear Model with Kernels (PLM-NW)

Input: Data matrix $[\mathbf{D}, \mathbf{Z}]$, label $\mathbf{y}$
Estimation of β:
1. fit $\mathbf{y} \sim \mathbf{Z}$ via kernel regression to derive $\mathbb{E}(\mathbf{y}|\mathbf{Z})$;
2. fit $\mathbf{D} \sim \mathbf{Z}$ via kernel regression to derive $\mathbb{E}(\mathbf{D}|\mathbf{Z})$;
3. fit $\mathbf{y} - \mathbb{E}(\mathbf{y}|\mathbf{Z}) \sim \mathbf{D} - \mathbb{E}(\mathbf{D}|\mathbf{Z})$ via OLS to derive $\hat{\boldsymbol{\beta}}$;

Estimation of f and Prediction of y:
4. fit $\mathbf{y} - \mathbf{D}\hat{\boldsymbol{\beta}} \sim \mathbf{Z}$ via kernel regression to derive $\hat{f}$ and define $\hat{\mathbf{y}} = \mathbf{D}\hat{\boldsymbol{\beta}} + \hat{f}(\mathbf{Z})$.

The kernel regression estimates the conditional expectation as a locally weighted average, using a kernel as a weighting function. For example, one may use the Nadaraya–Watson (NW) estimator $\widehat{m}_y(\mathbf{z})$ to learn $\mathbb{E}(\mathbf{y}|\mathbf{Z})$ as follows:

$$\widehat{m}_y(\mathbf{z}) = \frac{\sum_{i=1}^n \psi\left(\frac{\mathbf{z}-\mathbf{Z}i}{h_y}\right) y_i}{\sum_{i=1}^n \psi\left(\frac{\mathbf{z}-\mathbf{Z_i}}{h_y}\right)} \tag{A.1}$$

where h_y is a bandwidth whose size is related to the dependence of $\mathbf{y}$ on $\mathbf{Z}$ and $\psi(\cdot)$ is the kernel. Popular choices of kernels include uniform, triangle, Epanechnikov, Gaussian, quadratic and cosine. We use the Gaussian kernel, which is also known as the radial basis function (RBF) kernel, throughout the paper.

The second model is the DML with sample-splitting. We take a K-fold random partition $\{I_j\}_{j=1}^K$ of the indices $[n]$.

Algorithm 4 Double/Debiased Machine Learning (DML)

Input: Data matrix $[\mathbf{D}, \mathbf{Z}]$, label $\mathbf{y}$
for $j \in [K]$ **do**
1. fit $\mathbf{y}_{I_j}^C \sim \mathbf{Z}_{I_j}^C$ via some machine learning method to learn $\mathbb{E}(\mathbf{y}|\mathbf{Z})$;
2. fit $\mathbf{D}_{I_j}^C \sim \mathbf{Z}_{I_j}^C$ via some machine learning method to learn $\mathbb{E}(\mathbf{D}|\mathbf{Z})$;
3. fit $\mathbf{y}_{I_j} - \mathbb{E}(\mathbf{y}_{I_j}|\mathbf{Z}_{I_j}) \sim \mathbf{D}_{I_j} - \mathbb{E}(\mathbf{D}_{I_j}|\mathbf{Z}_{I_j})$ via OLS to derive $\hat{\boldsymbol{\beta}}$, denoted as $\hat{\boldsymbol{\beta}}^{(j)}$;

end for
4. aggregate the estimators: $\hat{\boldsymbol{\beta}} = \sum_j \hat{\boldsymbol{\beta}}^{(j)}/K$.

We note that the DML does not explicitly predict $\mathbf{y}$, hence we add an extra step to accomplish this. Denote the estimators in step 1 and step 2 as $m_y^{(j)}$ and $m_D^{(j)}$. We aggregate the estimates of $\mathbb{E}(\mathbf{y}|\mathbf{Z})$ and $\mathbb{E}(\mathbf{D}|\mathbf{Z})$ from the K estimators and predict

$$\hat{\mathbf{y}} := \sum_j m_y^{(j)}(\mathbf{Z})/K + \left(\mathbf{D} - \sum_j m_D^{(j)}(\mathbf{Z})/K\right)\hat{\boldsymbol{\beta}}.$$

We remark that the choice of the machine learning methods to use is flexible. For instance, one may use Lasso if $\mathbf{Z}$ is high dimensional or use the logistic regression if $\mathbf{D}$ is categorical.

B Proof of Theorem 1

Denote label as $\mathbf{y} \in \mathbb{R}^n$, data as $\mathbf{X} \in \mathbb{R}^{n\times p}$, the features selected by Lasso as $\mathbf{D} \in \mathbb{R}^{n\times p_L}$ and the rest as $\mathbf{Z} \in \mathbb{R}^{n\times p_N}$, where $p_L + p_N = p$. Recall that $\mathbf{W} \in \mathbb{R}^{(p_N)\times m}, \mathbf{A} \in \mathbb{R}^{m\times(1+p_L)}$ are the weights in first and second layers respectively. Here we consider a neural network of the following form.

$$F(\mathbf{W}, \mathbf{A}, \mathbf{z}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} \mathbf{A}_r \sigma\left(\mathbf{w}_r^\top \mathbf{z}\right)$$

where $\mathbf{w}_r$ and $\mathbf{A}_r$ are the weights corresponding to the r-th neuron in the hidden layer. The input $\mathbf{z} \in \mathbf{R}^{p_N}$ and $\sigma(\cdot)$ is the ReLU activation function. To be more clear, we note the s-th dimension of F as $F_s(\mathbf{W}, \mathbf{A}, \mathbf{z})$, which means for $s = 1, 2, \cdots, 1 + p_L$,

$$F_s(\mathbf{W}, \mathbf{A}, \mathbf{z}) = [F(\mathbf{W}, \mathbf{A}, \mathbf{z})]_s = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} A_{rs} \sigma\left(\mathbf{w}_r^\top \mathbf{z}\right) \tag{B.1}$$

Here A_{rs} is a scalar representing the output weight in the second layer. Given the dataset $\{(\mathbf{Z}_i, \mathbf{M}_i)\}_{i=1}^n$ with the multivariate response $\mathbf{M} := [\mathbf{y}, \mathbf{D}]$. We aim to minimize

$$L(\mathbf{W}, \mathbf{A}) = \sum_{i=1}^{n} \frac{1}{2} \left\| F(\mathbf{W}, \mathbf{A}, \mathbf{Z}_i) - \mathbf{M}_i \right\|_2^2 .$$

Formally, we consider the gradient flow of the gradient descent defined by:

$$\frac{d\mathbf{w}_r(t)}{dt} = -\frac{\partial L(\mathbf{W}(t), \mathbf{A})}{\partial \mathbf{w}_r(t)}$$

for $r = 1, \ldots, m$. Simple chain rule gives the MSE loss derivative with respect to each weight vector $\mathbf{w_r}$ as

$$\begin{aligned}
\frac{\partial L(\mathbf{W}, \mathbf{A})}{\partial \mathbf{w}_r} &= \sum_{i=1}^{n} \sum_{s=1}^{(1+p_L)} (F_s(\mathbf{W}, \mathbf{A}, \mathbf{Z}_i) - M_{is}) \frac{\partial F_s(\mathbf{W}, \mathbf{A}, \mathbf{Z}_i)}{\partial \mathbf{w}_r} \\
&= \frac{1}{\sqrt{m}} \sum_{i=1}^{n} \sum_{s=1}^{(1+p_L)} (F_s(\mathbf{W}, \mathbf{A}, \mathbf{Z}_i) - M_{is}) \mathbf{A}_{rs} \mathbf{Z}_i \mathbb{I}\left\{\mathbf{w}_r^\top \mathbf{Z}_i \geq 0\right\},
\end{aligned} \tag{B.2}$$

as the form in (B.1) indicates

$$\frac{\partial F_s(\mathbf{W}(t), \mathbf{A}, \mathbf{Z}_i)}{\partial \mathbf{w}_r(t)} = \frac{1}{\sqrt{m}} \mathbf{A}_{rs} \mathbf{Z}_i \mathbb{I}\left\{\mathbf{w}_r^\top \mathbf{Z}_i \geq 0\right\} \tag{B.3}$$

Let us shorthand $u_{is}(t) = F_s(W(t), A, Z_i)$. The dynamics of each dimension of one prediction is again given by the chain rule,

$$\begin{aligned}
\frac{d}{dt} u_{is}(t) =& \sum_{r=1}^{m} \langle \frac{\partial F_s(\mathbf{W}(t), \mathbf{A}, \mathbf{Z}_i)}{\partial \mathbf{w}_r(t)}, \frac{d\mathbf{w}_r(t)}{dt} \rangle \\
=& \sum_{r=1}^{m} \langle \frac{\partial F_s(\mathbf{W}, \mathbf{A}, \mathbf{Z}_i)}{\partial \mathbf{w}_r}, \sum_{j=1}^{n} \sum_{h=1}^{(1+p_L)} (M_{jh} - F_h(\mathbf{W}, \mathbf{A}, \mathbf{Z}_j)) \frac{\partial F_h(\mathbf{W}, \mathbf{A}, \mathbf{Z}_j)}{\partial \mathbf{w}_r} \rangle \\
=& \sum_{h=1}^{(1+p_L)} \sum_{j=1}^{n} (M_{jh} - F_h(\mathbf{W}, \mathbf{A}, \mathbf{Z}_j)) \sum_{r=1}^{m} \langle \frac{\partial F_s(\mathbf{W}, \mathbf{A}, \mathbf{Z}_i)}{\partial \mathbf{w}_r}, \frac{\partial F_h(\mathbf{W}, \mathbf{A}, \mathbf{Z}_j)}{\partial \mathbf{w}_r} \rangle \\
=& \sum_{h=1}^{(1+p_L)} \sum_{j=1}^{n} (M_{jh} - u_{jh}) (\mathbf{H}_{sh})_{ij}(t).
\end{aligned} \tag{B.4}$$

Here we define the $n \times n$ matrix $\mathbf{H}_{sh}(t)$ using (B.3) as follows.

$$\begin{aligned}(\mathbf{H}_{sh})_{ij}(t) =& \sum_{r=1}^{m} \langle \frac{\partial F_s(\mathbf{W}, \mathbf{A}, \mathbf{Z}_i)}{\partial \mathbf{w}_r}, \frac{\partial F_h(\mathbf{W}, \mathbf{A}, \mathbf{Z}_j)}{\partial \mathbf{w}_r} \rangle \\ =& \frac{1}{m} \mathbf{Z}_i^\top \mathbf{Z}_j \sum_{r=1}^{m} A_{rs} A_{rh} \mathbb{I}\left\{ \mathbf{Z}_i^\top \mathbf{w}_r(t) \geq 0, \mathbf{Z}_j^\top \mathbf{w}_r(t) \geq 0 \right\}.\end{aligned}$$

Now we can write the dynamics of the predictions (B.4) in a compact way:

$$\begin{aligned}\frac{d}{dt}\mathbf{u}_s(t) =& \sum_{h=1}^{(1+p_L)} \mathbf{H}_{sh}(t)(\mathbf{M}_s - \mathbf{u}_s(t)) \\ \frac{d}{dt}(\mathbf{M}_s - \mathbf{u}_s(t)) =& -\sum_{h=1}^{(1+p_L)} \mathbf{H}_{sh}(t)(\mathbf{M}_s - \mathbf{u}_s(t))\end{aligned} \tag{B.5}$$

Furthermore, we can rewrite (B.5) by concatenating each dimension of the prediction sequentially and denote the concatenated response in $\mathbb{R}^{n(1+p_L)\times 1}$ as $\mathbf{M}_{conc} - \mathbf{u}_{conc}(t)$. Then the dynamics is equivalent to

$$\frac{d}{dt}(\mathbf{M}_{conc} - \mathbf{u}_{conc}(t)) = \mathbf{H}_{whole}(\mathbf{M}_{conc} - \mathbf{u}_{conc}(t))$$

with

$$\mathbf{H}_{whole} := \begin{pmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} & \cdots & \mathbf{H}_{1,1+p_L} \\ \mathbf{H}_{21} & \mathbf{H}_{22} & \cdots & \mathbf{H}_{2,1+p_L} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{1+p_L,1} & \mathbf{H}_{1+p_L,2} & \cdots & \mathbf{H}_{1+p_L,1+p_L} \end{pmatrix}.$$

We now consider the NTK matrices corresponding to infinite-width neural network:

$$(\mathbf{H}_{sh})^\infty := \lim_{m\to\infty} (\mathbf{H}_{sh}) \quad \text{and} \quad \mathbf{H}_{whole}^\infty := \lim_{m\to\infty} \mathbf{H}_{whole}.$$

We notice that if $s = h$, since A_{rs} is ± 1, $(\mathbf{H}_{sh})^\infty$ is the same as $\mathbf{H}^\infty$ in Fact 4.1, which has been proven to be positive definite. If $s \neq h$, we have

$$\begin{aligned}(\mathbf{H}_{sh})_{ij}^\infty =& \mathbb{E}_{\mathbf{A}_s\sim\text{unif}\{-1,1\},\mathbf{A}_h\sim\text{unif}\{-1,1\},\mathbf{w}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}(\mathbf{Z}_i^\top \mathbf{Z}_j \mathbf{A}_s \mathbf{A}_h \mathbb{I}\left\{\mathbf{w}^\top \mathbf{Z}_i \geq 0, \mathbf{w}^\top \mathbf{Z}_j \geq 0\right\}) \\ =& \mathbf{Z}_i^\top \mathbf{Z}_j \mathbb{E}(\mathbf{A}_s)\mathbb{E}(\mathbf{A}_h)\mathbb{P}_{\mathbf{w}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left\{\mathbf{w}^\top \mathbf{Z}_i \geq 0, \mathbf{w}^\top \mathbf{Z}_j \geq 0\right\}) = 0\end{aligned}$$

From the initialization of A_{rs}, we get $\frac{1}{m}\sum_{r=1}^m A_{rs} \to \mathbb{E}(\mathbf{A}_s) = 0$ as $m \to \infty$ by the law of large numbers. Hence we obtain

$$\mathbf{H}_{whole}^\infty := \begin{pmatrix} \mathbf{H}^\infty & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{H}^\infty & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{H}^\infty \end{pmatrix}.$$

In summary, we conclude that $\lambda_0 := \lambda_{\min}(\mathbf{H}^\infty) = \lambda_{\min}(\mathbf{H}_{ss}^\infty) > 0$ and $\lambda_{\min}(\mathbf{H}_{sh}^\infty) = 0$ for $s \neq h$. Note that the eigenvalues of the block diagonal matrix $\mathbf{H}_{whole}^\infty$ is the same as those of $\mathbf{H}^\infty$, hence $\lambda_{\min}(\mathbf{H}_{whole}^\infty) = \lambda_0$.

To prove Theorem 1, we first show that $\mathbf{H}_{whole}(0)$ is close to $\mathbf{H}_{whole}^\infty$ and hence is positive definite.

Lemma B.1. *If* $m = \Omega\left(\frac{n^2(1+p_L)^2}{\lambda_0^2}\log\left(\frac{n^2(1+p_L)^2}{\delta}\right)\right)$, *then we have* $\|\mathbf{H}_{whole}(0) - \mathbf{H}_{whole}^\infty\|_2 \leq \frac{\lambda_0}{4}$ *and* $\lambda_{\min}(\mathbf{H}_{whole}(0)) \geq \frac{3}{4}\lambda_0$ *with probability of at least* $1 - \delta$.

Proof. By Hoeffding inequality, we have for each fixed (i,j) pair, we have with probability $1-\delta'$,

$$|\mathbf{H}_{whole}(0) - \mathbf{H}^{\infty}_{whole}| \leq \sqrt{\frac{2}{m}\log\frac{2}{\delta'}}$$

For all (i,j), if we set $\delta = n^2(1+p_L)^2\delta'$, then

$$|\mathbf{H}_{whole}(0) - \mathbf{H}^{\infty}_{whole}| \leq \sqrt{\frac{2}{m}\log\frac{2n^2(1+p_L)^2}{\delta}}$$

Hence, if $m = \Omega\left(\frac{n^2(1+p_L)^2}{\lambda_0^2}\log\left(\frac{n^2(1+p_L)^2}{\delta}\right)\right)$, then

$$\begin{aligned}\|\mathbf{H}_{whole}(0) - \mathbf{H}^{\infty}_{whole}\|_2^2 \leq& \|\mathbf{H}_{whole}(0) - \mathbf{H}^{\infty}_{whole}\|_F^2\\ \leq& \sum_{i,j}\left|(\mathbf{H}_{whole})_{ij}(0) - (\mathbf{H}_{whole})^{\infty}_{ij}\right|^2\\ \leq& \frac{2n^2(1+p_L)^2\log(2n^2(1+p_L)^2/\delta)}{m}\end{aligned}$$

which leads to

$$\|\mathbf{H}_{whole}(0) - \mathbf{H}^{\infty}_{whole}\|_2 \leq \frac{\lambda_0}{4}.$$

By the standard matrix concentration bound used in [15], $\mathbf{H}_{whole}(0)$ has a positive least eigenvalue with high probability:

$$\lambda_{\min}(\mathbf{H}_{whole}(0)) \geq \lambda_{\min}(\mathbf{H}^{\infty}_{whole}) - \frac{\lambda_0}{4}.$$

□

Lemma B.2. *If $\mathbf{w}_r$ are i.i.d generated from $\mathcal{N}(\mathbf{0},\mathbf{I})$ for $r\in[m]$, and $\|\mathbf{w}_r(0)-\mathbf{w}_r\|_2 \leq \frac{c\delta\lambda_0}{n^2(1+p_L)^2} =: R$ for some small positive constant c, then the following holds with probability at least $1-\delta$ we have $\|\mathbf{H}_{whole}-\mathbf{H}_{whole}(0)\|_2 < \frac{\lambda_0}{4}$ and $\lambda_{\min}(\mathbf{H}_{whole}) > \frac{\lambda_0}{2}$.*

Proof. First we set

$$A_r^i = \left\{\exists \mathbf{w} : \|\mathbf{w}-\mathbf{w}_r(0)\| \leq R, \mathbb{I}\left\{\mathbf{x}_i^\top \mathbf{w}_r(0)\geq 0\right\} \neq \mathbb{I}\left\{\mathbf{x}_i^\top \mathbf{w}\geq 0\right\}\right\}$$

Since $\mathbf{w}_r \sim N(\mathbf{0},\mathbf{I})$, By the anti-concentration inequality of Gaussian we have

$$P\left(A_r^i\right) = P_{w\sim N(0,1)}(|w|<R) \leq \frac{2R}{\sqrt{2\pi}}$$

Therefore, for any $\mathbf{w}_r$ that satisfies the assumption, we have

$$\begin{aligned}&\mathbb{E}\left[|(\mathbf{H}_{sh})_{ij}(0) - (\mathbf{H}_{sh})_{ij}|\right]\\ =&\mathbb{E}\left[\frac{1}{m}\left|\mathbf{Z}_i^\top\mathbf{Z}_j A_{rs}A_{rh}\sum_{r=1}^m\left(\mathbb{I}\left\{\mathbf{w}_r(0)^\top\mathbf{Z}_i\geq 0, \mathbf{w}_r(0)^\top\mathbf{Z}_j\geq 0\right\} - \mathbb{I}\left\{\mathbf{w}_r^\top\mathbf{Z}_i\geq 0, \mathbf{w}_r^\top\mathbf{Z}_j\geq 0\right\}\right)\right.\right].\\ \leq&\frac{1}{m}\sum_{r=1}^m\mathbb{E}\left[\mathbb{I}\left\{A_r^i\cup A_r^j\right\}\right] \leq \frac{4R}{\sqrt{2\pi}}\end{aligned}$$

Taking the sum over (i,j) we obtain

$$\mathbb{E}\left(\Sigma_{(i,j)=(1,1)}^{(n(1+p_L),n(1+p_L))}\|(\mathbf{H}_{sh})_{ij} - (\mathbf{H}_{sh})_{ij}(0)\|\right) \leq \frac{4(n(1+p_L))^2R}{\sqrt{2\pi}}$$

Furthermore, by Markov's inequality

$$\mathbb{E}\left(\Sigma_{(i,j)=(1,1)}^{(n(1+p_L),n(1+p_L))}\|(\mathbf{H}_{sh})_{ij} - (\mathbf{H}_{sh})_{ij}(0)\|\right) \leq \frac{4(n(1+p_L))^2R}{\sqrt{2\pi}\delta}$$

with probability $1-\delta$. By matrix perturbation theory we get the bound of $\mathbf{H}_{whole}$ with initialization,

$$\begin{aligned}\|\mathbf{H}_{whole}-\mathbf{H}_{whole}(0)\|_2 \leq & \|\mathbf{H}_{whole}-\mathbf{H}_{whole}(0)\|_F \\ \leq & \sum_{(s,h)=(1,1)}^{(1+p_L,1+p_L)} \sum_{(i,j)=(1,1)}^{(n,n)} |(\mathbf{H}_{sh})_{ij}-(\mathbf{H}_{sh})_{ij}(0)| \\ \leq & \frac{4n^2(1+p_L)^2R}{\sqrt{2\pi}\delta}\end{aligned}$$

Plugging in R,

$$\lambda_{\min}(\mathbf{H}_{whole}) \geq \lambda_{\min}(\mathbf{H}_{whole}(0)) - \frac{4n^2(1+p_L)^2R}{\sqrt{2\pi}\delta} > \frac{\lambda_0}{2}$$

□

Lemma B.3. *Assume $0 \leq t_1 \leq t$ and $\lambda_{\min}(\mathbf{H}_{whole}(t_1)) \geq \frac{\lambda_0}{2}$. Then we have $\|\mathbf{M}_s-\mathbf{u}_s(t)\|_2^2 \leq \exp(-\lambda_0(1+p_L)t)\|\mathbf{M}_s-\mathbf{u}_s(0)\|_2^2$ and $\|\mathbf{w}_r(t)-\mathbf{w}_r(0)\|_2 \leq \frac{\sqrt{n}}{\sqrt{m}\lambda_0}\sum_{h=1}^{1+p_L}\|\mathbf{M}_h-\mathbf{u}_h(0)\|_2 =: R'$.*

Proof.

$$\frac{d}{dt}(\mathbf{M}_{conc}-\mathbf{u}_{conc}(t)) = \mathbf{H}_{whole}(\mathbf{M}_{conc}-\mathbf{u}_{conc}(t))$$

Hence,

$$\begin{aligned}\frac{d}{dt}\|\mathbf{M}_{conc}-\mathbf{u}_{conc}(t)\|_2^2 = & -2(\mathbf{M}_{conc}-\mathbf{u}_{conc}(t))^\top \mathbf{H}_{whole}(\mathbf{M}_{conc}-\mathbf{u}_{conc}(t)) \\ \leq & -\lambda_0\|\mathbf{M}_{conc}-\mathbf{u}_{conc}\|_2^2.\end{aligned}$$

Therefore, we can bound the loss

$$\|\mathbf{M}_{conc}-\mathbf{u}_{conc}(t)\|_2^2 \leq \exp(-\lambda_0 t)\|\mathbf{M}_{conc}-\mathbf{u}_{conc}(0)\|_2^2.$$

Hence for each s,

$$\|\mathbf{M}_s-\mathbf{u}_s(t)\|_2^2 \leq \exp(-\lambda_0 t)\|\mathbf{M}_s-\mathbf{u}_s(0)\|_2^2,$$

which proves the exponentially fast convergence of each dimension and that all dimensions evolve under the same dynamics.

For $0 \leq t_1 \leq t$,

$$\begin{aligned}\left\|\frac{d}{dt_1}\mathbf{w}_r(t_1)\right\|_2 = & \left\|\frac{1}{\sqrt{m}}\sum_{i=1}^{n}\sum_{h=1}^{1+p_L}(M_{ih}-u_{ih})A_{rs}\mathbf{Z}_i\mathbb{I}\left\{\mathbf{w}_r(t_1)^\top\mathbf{Z}_i \geq 0\right\}\right\|_2 \\ \leq & \frac{1}{\sqrt{m}}\sum_{i=1}^{n}\sum_{h=1}^{1+p_L}|M_{ih}-u_{ih}(t_1)| \\ \leq & \frac{\sqrt{n}}{\sqrt{m}}\sum_{h=1}^{1+p_L}\|\mathbf{M}_h-\mathbf{u}_h(t_1)\|_2 \\ \leq & \frac{\sqrt{n}}{\sqrt{m}}\sum_{h=1}^{1+p_L}\exp\left(-\lambda_0 t_1/2\right)\|\mathbf{M}_h-\mathbf{u}_h(0)\|_2\end{aligned}$$

In the end,

$$\|\mathbf{w}_r(t)-\mathbf{w}_r(0)\|_2 \leq \int_0^t \left\|\frac{d}{ds}\mathbf{w}_r(s)\right\|_2 ds \leq \frac{\sqrt{n}}{\sqrt{m}\lambda_0}\sum_{h=1}^{1+p_L}\|\mathbf{M}_h-\mathbf{u}_h(0)\|_2$$

□

Lemma B.4. *If $R' < R$, then for all $t \geq 0$, we have $\lambda_{\min}(\mathbf{H}_{whole}(t)) \geq \frac{1}{2}\lambda_0$, for all $r \in [m]$, $\|\mathbf{w}_r(t) - \mathbf{w}_r(0)\|_2 \leq R'$ and for all $s \in [1 + p_L]$, $\|\mathbf{M}_s - \mathbf{u}_s(t)\|_2^2 \leq \exp(-\lambda_0 t)\|\mathbf{M}_s - \mathbf{u}_s(0)\|_2^2$.*

Proof. This lemma takes the same form as [15, Lemma 3.4]: given Lemma B.2 and Lemma B.3, the result clearly follows and we refer the interested readers to [15] for details. □

It is sufficient to show that $R' < R$ is equivalent to $m = \Omega\left(\frac{n^5 \sum_{h=1}^{1+p_L} \|\mathbf{M}_h - \mathbf{u}_h(0)\|_2^2}{\lambda_0^4 \delta^2}\right)$. We bound

$$\begin{aligned}
&\mathbb{E}\left[\|\mathbf{M}_s - \mathbf{u}_s(0)\|_2^2\right] \\
=&\sum_{i=1}^{n}\left(M_{is}^2 + M_{is}\mathbb{E}\left[F_s\left(\mathbf{W}(0), \mathbf{A}, \mathbf{Z}_i\right)\right] + \mathbb{E}\left[F_s\left(\mathbf{W}(0), \mathbf{A}, \mathbf{Z}_i\right)^2\right]\right) \\
=&\sum_{i=1}^{n}\left(M_{is}^2 + 1\right) = O(n)
\end{aligned}$$

Using the Markov's inequality, we have $\|\mathbf{M}_s - \mathbf{u}_s(0)\|_2^2 = O(\frac{n}{\delta})$ with probability at least $1 - \delta$. Hence we have $m = \Omega\left(\frac{(1+p_L)^5 n^6}{\delta^3 \lambda_0^4}\right)$.

C Proof of Theorem 2

Proof. Recall that in (2.3) we have

$$\mathcal{Y} = \mathbf{y} - \mathbb{E}(\mathbf{y}|\mathbf{Z}) = (\mathbf{D} - \mathbb{E}(\mathbf{D}|\mathbf{Z}))\boldsymbol{\beta} + \boldsymbol{\epsilon} = \mathcal{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}.$$

Unfortunately we only observe data with noises

$$\widetilde{\mathcal{Y}} = \mathcal{Y} + \boldsymbol{\epsilon}_Y = \mathcal{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} + \boldsymbol{\epsilon}_Y = \widetilde{\mathcal{X}}\boldsymbol{\beta} - \boldsymbol{\epsilon}_X\boldsymbol{\beta} + \boldsymbol{\epsilon} + \boldsymbol{\epsilon}_Y$$

Hence we can derive (6.1) as the following:

$$\begin{aligned}
\tilde{\boldsymbol{\beta}} =&(\widetilde{\mathcal{X}}^\top\widetilde{\mathcal{X}})^{-1}\widetilde{\mathcal{X}}^\top\widetilde{\mathcal{Y}} \\
=&(\widetilde{\mathcal{X}}^\top\widetilde{\mathcal{X}})^{-1}\widetilde{\mathcal{X}}^\top\left(\widetilde{\mathcal{X}}\boldsymbol{\beta} - \boldsymbol{\epsilon}_X\boldsymbol{\beta} + \boldsymbol{\epsilon} + \boldsymbol{\epsilon}_Y\right) \\
=&\boldsymbol{\beta} - (\widetilde{\mathcal{X}}^\top\widetilde{\mathcal{X}})^{-1}\widetilde{\mathcal{X}}^\top\boldsymbol{\epsilon}_X\boldsymbol{\beta} + (\widetilde{\mathcal{X}}^\top\widetilde{\mathcal{X}})^{-1}\widetilde{\mathcal{X}}^\top(\boldsymbol{\epsilon} + \boldsymbol{\epsilon}_Y) \\
=&\left(1 - (\widetilde{\mathcal{X}}^\top\widetilde{\mathcal{X}})^{-1}\widetilde{\mathcal{X}}^\top\boldsymbol{\epsilon}_X\right)\boldsymbol{\beta} + (\widetilde{\mathcal{X}}^\top\widetilde{\mathcal{X}})^{-1}\widetilde{\mathcal{X}}^\top(\boldsymbol{\epsilon}_Y + \boldsymbol{\epsilon}) \\
=&\left(1 - \left(\frac{\widetilde{\mathcal{X}}^\top\widetilde{\mathcal{X}}}{n}\right)^{-1}\frac{\widetilde{\mathcal{X}}^\top\boldsymbol{\epsilon}_X}{n}\right)\boldsymbol{\beta} + \left(\frac{\widetilde{\mathcal{X}}^\top\widetilde{\mathcal{X}}}{n}\right)^{-1}\frac{\widetilde{\mathcal{X}}^\top(\boldsymbol{\epsilon}_Y + \boldsymbol{\epsilon})}{n}.
\end{aligned}$$

We start with investigating the bias. Since $\boldsymbol{\epsilon}, \boldsymbol{\epsilon}_Y$ and $\boldsymbol{\epsilon}_X$ are independent of other variables, we have

$$\frac{\widetilde{\mathcal{X}}^\top(\boldsymbol{\epsilon}_Y + \boldsymbol{\epsilon})}{n} \to 0.$$

In addition,

$$\frac{\widetilde{\mathcal{X}}^\top\boldsymbol{\epsilon}_X}{n} = \frac{\mathcal{X}^\top\boldsymbol{\epsilon}_X + \|\boldsymbol{\epsilon}_X\|_2^2}{n} \to \mathbb{E}(\epsilon_X^2) = \sigma_X^2\mathbf{I}_{p_L}.$$

Next, we denote the convergence in probability as plim and look at

$$\frac{\widetilde{\mathcal{X}}^\top\widetilde{\mathcal{X}}}{n} \to \operatorname{plim}\frac{\mathcal{X}^\top\mathcal{X}}{n} + \operatorname{plim}\frac{\boldsymbol{\epsilon}_X^\top\boldsymbol{\epsilon}_X}{n} = \mathcal{Q} + \sigma_X^2\mathbf{I}_{p_L}$$

where $\mathcal{Q} := \operatorname{plim}\frac{\mathcal{X}^\top\mathcal{X}}{n}$ exists by the law of large numbers because $\mathcal{X}$ is i.i.d. in rows. Therefore, we obtain

$$\widetilde{\boldsymbol{\beta}} \to \left(\mathbf{I} - \sigma_X^2\left(\mathcal{Q} + \sigma_X^2\mathbf{I}_{p_L}\right)^{-1}\right)\boldsymbol{\beta} := (\mathbf{I} - \mathbf{R})\boldsymbol{\beta}$$

where $\mathbf{R} := \sigma_X^2 \left(\mathcal{Q} + \sigma_X^2 \mathbf{I}_{p_L}\right)^{-1} = \sigma_X^2 \text{plim} \left(\frac{\widetilde{\mathcal{X}}^\top \widetilde{\mathcal{X}}}{n}\right)^{-1}$. We claim that $\widetilde{\boldsymbol{\beta}}$ is a consistent estimator of $\boldsymbol{\beta}$ if and only if m_D is consistently approximated (meaning $\sigma_X^2 = 0$). If m_D is not consistently approximated, we can modify the estimator via $\mathbf{R} \in \mathbb{R}^{p_L \times p_L}$ and the new estimator $(\mathbf{I} - \mathbf{R})^{-1} \widetilde{\boldsymbol{\beta}}$ which is a consistent estimator of $\boldsymbol{\beta}$.

To establish the $\sqrt{n}$-consistency, let us consider the asymptotic distribution of the OLS estimator. Multiplying $\sqrt{n}$ on $\widetilde{\boldsymbol{\beta}} - (\mathbf{I} - \mathbf{R})\boldsymbol{\beta}$ and taking to the limit, we have

$$\sqrt{n}\left(\widetilde{\boldsymbol{\beta}} - (\mathbf{I} - \mathbf{R})\boldsymbol{\beta}\right) \to \text{plim} \left(\frac{\widetilde{\mathcal{X}}^\top \widetilde{\mathcal{X}}}{n}\right)^{-1} \frac{\widetilde{\mathcal{X}}^\top (\boldsymbol{\epsilon}_Y + \boldsymbol{\epsilon})}{\sqrt{n}}$$

and

$$n\left(\widetilde{\boldsymbol{\beta}} - (\mathbf{I} - \mathbf{R})\boldsymbol{\beta}\right)\left(\widetilde{\boldsymbol{\beta}} - (\mathbf{I} - \mathbf{R})\boldsymbol{\beta}\right)^\top \to \text{plim} \left(\frac{\widetilde{\mathcal{X}}^\top \widetilde{\mathcal{X}}}{n}\right)^{-1} \left(\frac{\widetilde{\mathcal{X}}^\top (\boldsymbol{\epsilon}_Y + \boldsymbol{\epsilon})}{\sqrt{n}}\right)\left(\frac{\widetilde{\mathcal{X}}^\top (\boldsymbol{\epsilon}_Y + \boldsymbol{\epsilon})}{\sqrt{n}}\right)^\top \left(\frac{\widetilde{\mathcal{X}}^\top \widetilde{\mathcal{X}}}{n}\right)^{-1}$$

Making use of $\text{plim} \left(\frac{\widetilde{\mathcal{X}}^\top \widetilde{\mathcal{X}}}{n}\right)^{-1} = \mathbf{R}/\sigma_X^2$ and after some calculation, we arrive at

$$n\left(\widetilde{\boldsymbol{\beta}} - (\mathbf{I} - \mathbf{R})\boldsymbol{\beta}\right)\left(\widetilde{\boldsymbol{\beta}} - (\mathbf{I} - \mathbf{R})\boldsymbol{\beta}\right)^\top \to \frac{\left(\sigma_\epsilon^2 + \sigma_Y^2\right)\mathbf{R}}{\sigma_X^2}$$

Thus we complete the proof. In addition, the asymptotic normality can be easily derived by applying the central limit theorem and the Slutsky's theorem. □

D Details of Experiments and Extra Application

D.1 Data generation for Table 1

In Table 1 and Figure 2, we generate $\mathbf{D}$ and $\mathbf{y}$ using $\mathbf{Z}$, which is generated by a multivariate standard normal distribution:

$$\mathbf{D} = 50 \sum_{j=1}^{10} \sin \mathbf{Z}_j + \mathcal{N}(\mathbf{0}, \mathbf{I})$$
$$\mathbf{y} = \mathbf{D} + \sum_{j=1}^{10} \cosh \mathbf{Z}_j + \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Here $\mathbf{X} \in \mathbb{R}^{10000 \times 11} = [\mathbf{D}, \mathbf{Z}]$, $\mathbf{D} \in \mathbb{R}^{n \times 1}$, $\mathbf{Z} \in \mathbb{R}^{10000 \times 10}$. Though the problem here is non-linear (in fact partially linear), unlike in the debiasing setting in Table 2, it is fair since all other PLMs work on such problem. Our goal is to demonstrate that PLM-NN is a strong candidate in the PLM family. Here we consider a univaraite $\mathbf{D}$, so that DML can apply methods including vanilla Lasso to solve this problem. In addition, we design $\mathbf{D}$ that is dependent on $\mathbf{Z}$ so that the NW kernel can use a finite bandwidth.

For Table 1, we train the two-layer neural network with Adam, width 10000 and learning rate 0.0002. For and Figure 2, we train the same network with full-batch gradient descent and learning rate 0.01.

D.2 Data generation for Table 2

In Table 2 and Figure 3, we have

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \mathcal{N}(\mathbf{0}, \mathbf{I}) = \mathbf{D}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \mathcal{N}(\mathbf{0}, \mathbf{I})$$

where $\boldsymbol{\theta} = [1, 1, \ldots, 0]$ with the first k entries as ones and the rest as zeros. $\mathbf{X}$ is generated from a multivariate standard normal distribution. In the second equality, we have $\mathbf{D}$ as the columns from $\mathbf{X}$ that are selected by Lasso. The Lasso penalties are chosen specifically to select a moderate number of features so that OLS is available. Here $\mathbf{D} \in \mathbb{R}^{1000 \times k_{Lasso}}, \mathbf{Z} \in \mathbb{R}^{1000 \times (p - k_{Lasso})}$, with $k_{Lasso} = \#\{j : [\widehat{\boldsymbol{\beta}}_{Lasso}]_j \neq 0\}$. We note that when the dimension

is relatively high or when the sparsity is relatively large, it is impossible to achieve full power (or true positive rate, or precision), no matter how one carefully tunes the penalty of Lasso. In other words, Lasso must select in some null signals. This phenomenon, known as the Donoho-Tanner phase transition, motivates our experimental settings that consider high dimension and/or high sparsity. For Table 2 and Figure 3, we train the two-layer neural network with Adam, width 1000 and learning rate 0.0002.

D.3 Complementary Experiments to Table 1

In this section, we conduct experiments on more complicated synthetic data to complement Table 1 and Figure 4. Our new setting is $\mathbf{D} = \sin(\mathbf{T}_1) + \log(\mathbf{T}_2 + 1) + \frac{1}{1+\mathbf{T}_3} + \max(0, \mathbf{T}_4) + \mathbf{T}_5^2 + \mathcal{N}(0, \mathbf{I})$ and $\mathbf{y} = \mathbf{D} + \cosh(\mathbf{T}_1) + \mathbf{T}_2 + \mathbf{T}_3 \times \mathbf{T}_4 + \mathcal{N}(0, \mathbf{I})$. It is clear that PLM-NN again demonstrates high level of performance in terms of consistency and accuracy, with linear convergence in the training dynamics.

PLMs	Est MSE (10^{-5})	Train MSE	Test MSE
PLM-NN	4.00	3.81	3.87
PLM-NW	1.26	3.51	3.66
DML Lasso	0.38	4.17	3.99
DML DT	240	8.41	15.51

Table 4: Comparison of PLMs in 50 independent runs with new data generation. Here PLMNW denotes the PLM using NW kernel, DT denotes decision trees at depth of 2.

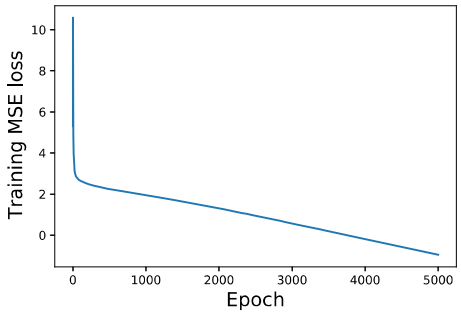

Figure 5: Same setting as Table 4 except $n = 100$ and $\mathbf{Z}$ is normalized. The loss is in logarithmic scale.

D.4 Adversarial attack on tabular data

Using the knowledge of the impact of a feature on the output, we can design adversarial samples to attack a potentially strong trained model. The dataset we use is Default of Credit Card Clients Dataset from UCI. The dataset includes the information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005. The whole dataset includes 25 columns and a sample size of 30000. We'll conduct null value imputation by association rules between categorical variables.

We preprocess the dataset in the following way: we use the known columns to train logistic regressions and to predict the missing values in the columns 'MARRIAGE' and 'EDUCATION'. Finally we conduct the standard scaling for $\mathbf{X}$ after one-hot encoding all the categorical columns.

We attack a ReLU activated MLP of three layers with the DebiNet and the traditional OLS. Our attacking mechanism takes the column with the maximum coefficient magnitude calculated by OLS or some feature selection methods. Once we find the targeted column, say the j-th column, we perturb the values towards the maximum or minimum value within the column, depending on the sign of $\hat{\beta}_j$.

The DebiNet is equipped with Lasso or Elastic net. The attacks show the same performance because the most influencing feature given by all three feature selection methods is identical. We note that the feature under

Methods	Val loss	Accuracy	AUC score
MLP Baseline	0.43	0.824	0.77
OLS Attack	0.39	0.840	0.69
DebiNet Attack	3.95	0.008	0.02

Table 5: Adversarial attack on tabular dataset: the Taiwan credit.

perturbation is the first repayment status, indicating that not paying back the first credit bill has a large impact on the credibility of a client. The perturbing columns selected by OLS and DebiNet are different. OLS fails to select what DebiNet selects because some features have infinity coefficients, which might caused by model overfitting and consequently large variances of the parameter estimators. The validation loss, accuracy and AUC score in Table 5 show that OLS attack is not effective as DebiNet.

E General Losses, Activation Functions and Other Optimizers

We demonstrate the convergence (in log-scale) of two-layer, fully connected, multivariate output neural networks with the same input distribution and weight initialization as in Figure 2. The baseline is the ReLU activation, full-batch gradient descent and MSE loss and we change only one element in the baseline at a time.

Similar to the univariate output case, we observe that with sufficiently wide hidden layer and sufficiently small learning rate, the neural networks may converge to the global minimum at a linear rate with different optimizers, losses and activation functions. In particular, we note that the performance in this section is not yet supported by theory and thus suggests that NTK theory may be richer than it currently is. We highlight that advanced tools from linear algebra are required to analyze the positive definiteness of the NTK matrices incurred by other activation functions. Moreover, different optimizers lead to different gradient flows and, together with different losses, to different matrix ordinary differential equations. It would be interesting to investigate the effects of different dynamics on the convergence under the NTK framework.

E.1 Optimizers besides Gradient Descent

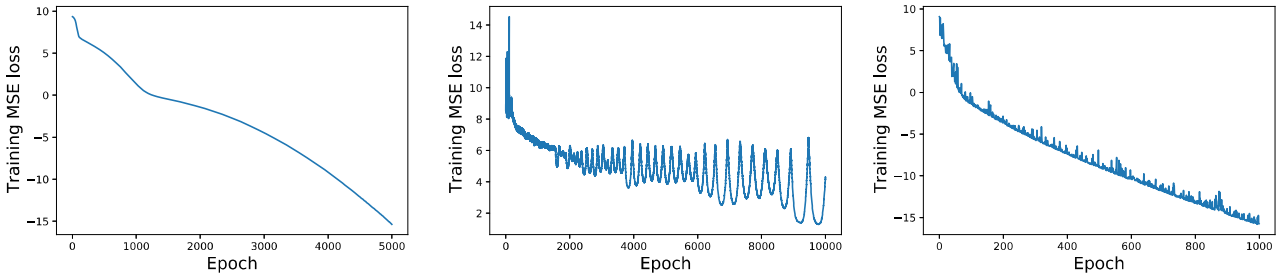

Figure 6: Left: Adam with learning rate 0.001 and full batch size. Middle: Nesterov-accelarated gradient descent with learning rate 0.01. Right: SGD with learning rate 0.01 and batch size 8.

In Figure 6, Adam converges to zero loss exponentially fast but the Nesterov-accelarated gradient descent seems not to. This shows that momentum has a significant impact on the convergence. It is also interesting to note that SGD converges much faster than the gradient descent, at a linear rate. The plot shows some fluctuations due to the randomness of sub-sampling and coincides with the dynamics described in the univariate output case.

E.2 General Losses and Activation Functions

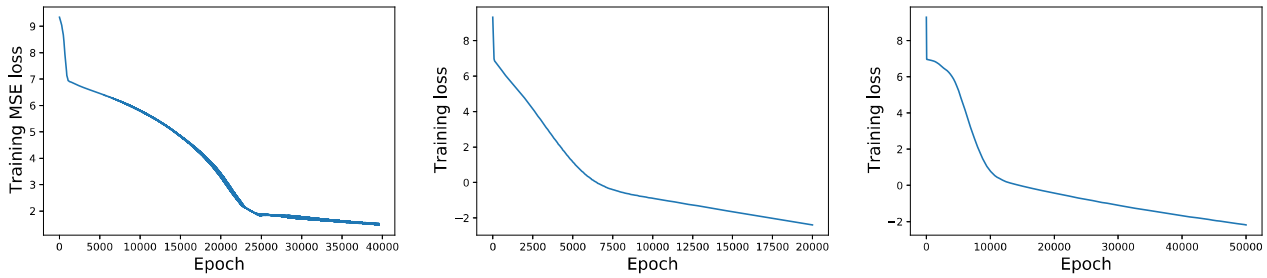

Figure 7: Left: Huber loss. Middle: Leaky ReLU activation. Right: Tanh activation.

In Figure 7, we empirically illustrate that loss functions play a key role in the training dynamics as they directly affect the dynamics. While the activation functions may have weaker effects on whether the dynamics converge to the global minimum, they can influence the rates at which the convergence takes place.