Supplement to: Automatic debiasing of neural networks via moment-constrained learning

Equation and reference numbering in this document continues from that of the main manuscript.

Appendix A. Notes on the numerical experiments

A.1. MADNet architecture

To ensure a fair evaluation, our proposed MADNet architecture emulates that of the RieszNet (Chernozhukov et al., 2022a) (see Figure 2 for a schematic of the multi-headed architecture). MADNet uses a shared network of width 200 and depth 3 followed by three branches: 2 outcome networks (one per binary treatment) each of width 100 and depth 2 and another of depth zero, i.e. a linear combination of the final shared representation layer that is our $\hat{\beta}_\perp$ prediction. The constraint weight hyperparameter was set to $\tilde{\lambda} = 5$, the weight mixing parameter was set to $\rho = 1$, and Exponential Linear Unit (ELU) activation functions were used throughout. Finally, outcomes Y were scaled by their sample standard deviation prior to training, with predictions rescaled to the original scale using the same constant standard deviation estimate.

Ablation study: We compared the performance of the MADNet proposal across learner architectures, by conducting an ablation study wherein the multi-headed architecture was replaced with a fully connected MLP architecture. In particular, we used a standard feed-forward network of width 200 and depth 4 along with the same hyperparameters outlined above. Results, reported in Table 2, indicate that the multi-headed architecture leads to a modest reduction in mean absolute error (MAE) in all but the MADNet (IPW) estimator, and that MADNet estimators tend to outperform their RieszNet counterparts using both architectures (in terms of reduced MAE).

Hyperparameter sensitivity: We examine sensitivity of our proposal to the penalization strength by running the MADNet with the $\tilde{\lambda} = 1$, and other parameters unchanged. Results, reported in Table 3 show slightly worse performance (increased MAE and increased Median absolute error), compared to results in Table 2, where $\tilde{\lambda} = 5$. This suggests that a high degree of weight should be given to satisfying the moment constraint.

A.2. MADNet training details

Numerical experiments were run on an Apple M2 Max chip with 32GB of RAM. The MADNet training procedure was also borrowed from Chernozhukov et al. (2022a, Appendix A1), which itself was borrowed from Shi et al. (2019). Minor modifications are outlined below. The dataset was split into a training dataset (80%) and validation dataset (20%), with estimation performed on the entire dataset. The training followed a two stage procedure outlined below.

ATE benchmarks

1. Fast training: batch size: 64, learning rate: 0.0001, maximum number of epochs: 100, optimizer: Adam, early stopping patience: 2, L2 weight decay: 0.001

2. Fine-tuning: batch size: 64, learning rate: 0.00001, maximum number of epochs: 600, optimizer: Adam, early stopping patience: 40, L2 weight decay: 0.001

ADE benchmarks

1. Fast-training: batch size: 64, learning rate: 0.001, maximum number of epochs: 100, optimizer: Adam, early stopping patience: 2, L2 weight decay: 0.001

2. Fine-tuning: batch size: 64, learning rate: 0.0001, maximum number of epochs: 300, optimizer: Adam, early stopping patience: 20, L2 weight decay: 0.001

The differences between the original implementations and ours are:

- For ADE moment estimation, RieszNet uses a finite difference approximation to differentiate the forward pass with respect to the treatment a. However our implementation uses automatic differentiation provided by JAX. One of the advantages of JAX is that the ADE can be straightforwardly expressed as `jax.grad(f)(a, x)`.
- On top of the early stopping callback, the original RieszNet and DragonNet implementations additionally use a learning rate plateau schedule that halves the learning rate when the validation loss metric has stopped improving over a short patience of epochs (shorter than the stopping patience). Whilst we implement the same two-stage training with early stopping, we use a constant learning rate in each of the fast-training and fine-tuning phases.
- L2 regularization is implemented differently between RieszNet and DragonNet. DragonNet use a regularizer to apply a penalty on the layer's kernel whilst RieszNet uses an additive L2 regularization term in their loss function (Chernozhukov et al., 2022a, Equation 5). However, recent work shows that L2 regularization and weight decay regularization are not equivalent for adaptive gradient algorithms, such as Adam (Loshchilov and Hutter, 2019). For this reason, we use Adam with weight decay regularization (provided by `optax.adamw`).
- We use a larger learning rate (0.9) for the constant additive bias parameters associated with the MLP outputs for the outcome, i.e. $f_{w,2}$ and $f_{w,3}$.

A.3. Naive Lagrangian optimization

We consider the basic differential multiplier method (BDMM), as described by Platt et al. (Platt and Barr, 1987). The authors introduce a so-called damping term $\delta \geq 0$ to the Lagrangian in (8) to obtain the Lagrangian

$$\mathcal{L}_\delta(f, \lambda) \equiv \mathbb{E}\left[\{\beta(Z) - f(Z)\}^2\right] + \lambda h(f) + \delta h^2(f),$$

with (8) recovered by setting $\delta = 0$. Note that when the moment constraint is satisfied, i.e.$h(f) = 0$, then $\mathcal{L}_\delta$ does not depend on δ. In Figure 4 we see how Naively performing gradient ascent on λ and gradient descent over f results in oscillatory behavior. Similar behavior is also observed in the literature on adversarial learning, see e.g. (Schäfer and Anandkumar, 2019; Mokhtari et al., 2020).

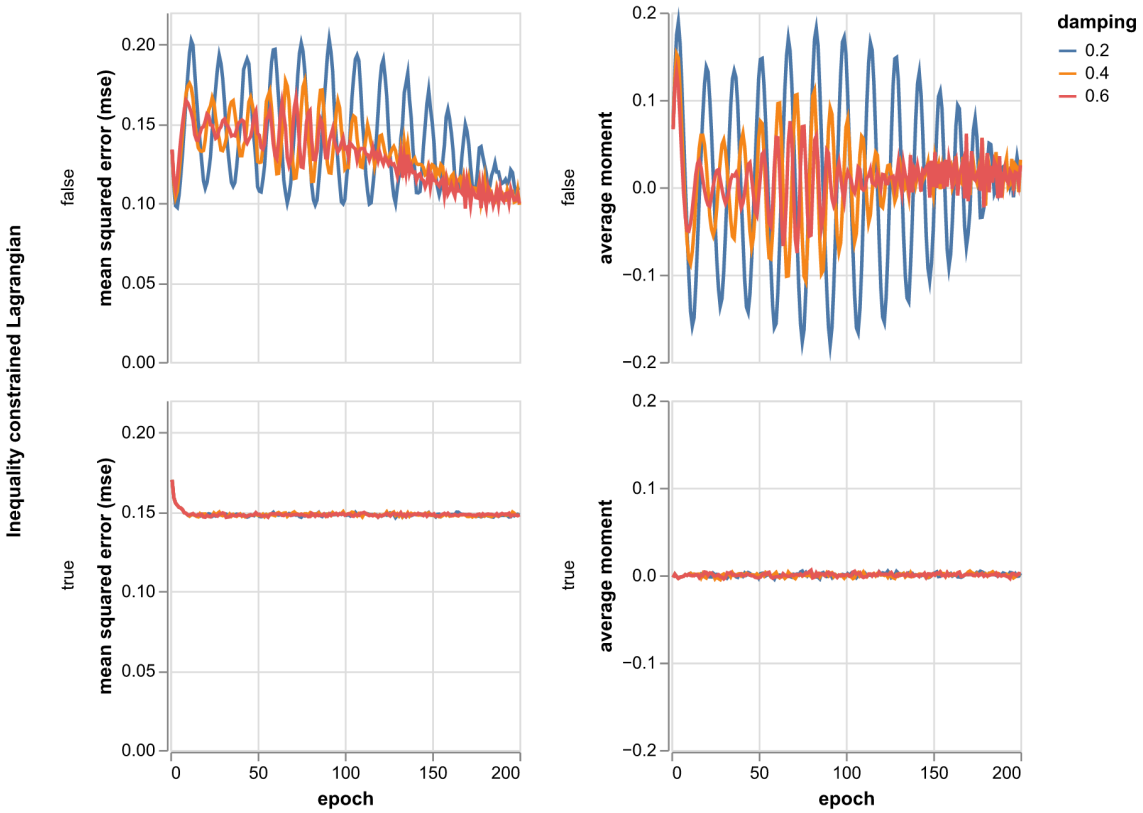

Figure 4: Top row: Low damping coefficients in the basic differential multiplier method (BDMM) (Platt and Barr, 1987) lead to oscillatory behavior around the saddle point solution when the optimisation problem is formulated as an equality constrained Lagrangian. Bottom row: Using the inequality constrained Lagrangian approach described in the main paper results in more stable training and constraint satisfaction. A single dataset from the IHDP data is used to showcase this behavior over 200 epochs.

Table 2: Full reproduction results for our own implementation of each learner/estimator. Here + SRR, refers to estimator which use the outcome model $\tilde{g}$ described in Chernozhukov et al. (2022a).

Dataset	Estimator	Architecture	Mean Absolute Error (MAE)	Median Absolute Error	Standard Error in MAE
BHP	MADNet (DR)	Fully connected	0.417	0.394	0.021
		Multiheaded	0.391	0.346	0.019
	MADNet (Direct)	Fully connected	0.512	0.427	0.029
		Multiheaded	0.471	0.424	0.026
	MADNet (IPW)	Fully connected	0.407	0.352	0.023
		Multiheaded	0.474	0.404	0.026
	RieszNet (DR + SRR)	Fully connected	0.447	0.370	0.024
		Multiheaded	0.428	0.355	0.023
	RieszNet (DR)	Fully connected	0.447	0.372	0.024
		Multiheaded	0.428	0.353	0.023
	RieszNet (Direct + SRR)	Fully connected	0.771	0.637	0.041
		Multiheaded	0.724	0.617	0.042
	RieszNet (Direct)	Fully connected	0.733	0.619	0.039
		Multiheaded	0.692	0.585	0.040
	RieszNet (IPW + SRR)	Fully connected	0.477	0.432	0.025
		Multiheaded	0.449	0.384	0.025
	RieszNet (IPW)	Fully connected	0.477	0.432	0.025
		Multiheaded	0.449	0.384	0.025
IHDP	DragonNet (DR + SRR)	Multiheaded	0.101	0.085	0.003
	DragonNet (DR)	Multiheaded	0.100	0.084	0.002
	DragonNet (Direct + SRR)	Multiheaded	0.124	0.098	0.004
	DragonNet (Direct)	Multiheaded	0.123	0.098	0.004
	DragonNet (IPW + SRR)	Multiheaded	0.262	0.233	0.006
	DragonNet (IPW)	Multiheaded	0.262	0.233	0.006
	MADNet (DR)	Fully connected	0.096	0.079	0.003
		Multiheaded	0.094	0.076	0.002
	MADNet (Direct)	Fully connected	0.527	0.383	0.018
		Multiheaded	0.504	0.367	0.016
	MADNet (IPW)	Fully connected	0.680	0.263	0.037
		Multiheaded	0.719	0.277	0.039
	RieszNet (DR + SRR)	Fully connected	0.119	0.091	0.004
		Multiheaded	0.109	0.088	0.003
	RieszNet (DR)	Fully connected	0.119	0.090	0.004
		Multiheaded	0.109	0.089	0.003
	RieszNet (Direct + SRR)	Fully connected	0.135	0.099	0.006
		Multiheaded	0.126	0.102	0.004
	RieszNet (Direct)	Fully connected	0.135	0.105	0.004
		Multiheaded	0.118	0.099	0.003
	RieszNet (IPW + SRR)	Fully connected	0.690	0.304	0.035
		Multiheaded	0.665	0.300	0.036
	RieszNet (IPW)	Fully connected	0.690	0.304	0.035
		Multiheaded	0.665	0.300	0.036

Appendix B. Short notes and proofs

B.1. First-order remainder under the standard theory

Claim: $\mathbb{E}[\hat{\varphi}(W) - \Psi] = -\langle \hat{\mu} - \mu, \hat{\alpha} - \alpha \rangle$ where $\hat{\varphi}(W) = m(\hat{\mu}, W) + \hat{\alpha}(Z)\{Y - \hat{\mu}(Z)\}$.

Table 3: Numerical experiment results for the Multiheaded MADNet procedure with $\tilde{\lambda} = 1$.

Dataset	Estimator	Mean Absolute Error (MAE)	Median Absolute Error	Standard Error in MAE
BHP	MADNet (DR)	0.407	0.370	0.021
	MADNet (Direct)	0.479	0.415	0.025
	MADNet (IPW)	0.544	0.459	0.031
IHDP	MADNet (DR)	0.098	0.077	0.003
	MADNet (Direct)	0.519	0.382	0.016
	MADNet (IPW)	0.712	0.283	0.038

Proof:

$$\begin{aligned}
&\mathbb{E}[m(\hat{\mu}, W) + \hat{\alpha}(Z)\{Y - \hat{\mu}(Z)\} - \Psi] \\
=&\mathbb{E}[m(\hat{\mu}, W) + \hat{\alpha}(Z)\{\mu(Z) - \hat{\mu}(Z)\} - m(\mu, W)] \\
=&\mathbb{E}[m(\hat{\mu} - \mu, W) - \hat{\alpha}(Z)\{\hat{\mu}(Z) - \mu(Z)\}] \\
=&\langle \hat{\mu} - \mu, \alpha \rangle - \langle \hat{\mu} - \mu, \hat{\alpha} \rangle \\
=&-\langle \hat{\mu} - \mu, \hat{\alpha} - \alpha \rangle.
\end{aligned}$$

B.2. Second-order remainder under the standard theory

Claim: If $\hat{\mu}$ and $\hat{\alpha}$ are consistent estimators for μ and α obtained from an independent sample, and there exists a constant M such that $\alpha^2(Z) < M$ and $\text{var}(Y|Z) < M$ almost surely, then $G_n[\hat{\varphi}(W) - \varphi(W)] = o_p(1)$.

Proof:

$$\begin{aligned}
G_n[\hat{\varphi}(W) - \varphi(W)] = &+ G_n[m(\hat{\mu} - \mu, W)] \\
&- G_n\left[\{\hat{\alpha}(Z) - \alpha(Z)\}\{\hat{\mu}(Z) - \mu(Z)\}\right] \\
&- G_n\left[\alpha(Z)\{\hat{\mu}(Z) - \mu(Z)\}\right] \\
&+ G_n\left[\{\hat{\alpha}(Z) - \alpha(Z)\}\{Y - \mu(Z)\}\right]
\end{aligned}$$

By the central limit theorem, these empirical processes are $o_p(1)$ when the following expressions are $o_p(1)$

$$\begin{aligned}
&\mathbb{E}\left[m^2(\hat{\mu} - \mu, W)\right] \\
&\mathbb{E}\left[\{\hat{\alpha}(Z) - \alpha(Z)\}^2\{\hat{\mu}(Z) - \mu(Z)\}^2\right] \\
&\mathbb{E}\left[\alpha^2(Z)\{\hat{\mu}(Z) - \mu(Z)\}^2\right] \\
&\mathbb{E}\left[\{\hat{\alpha}(Z) - \alpha(Z)\}^2\{Y - \mu(Z)\}^2\right]
\end{aligned}$$

The first two terms are $o_p(1)$ by consistency of $\hat{\alpha}$ and $\hat{\mu}$, for the final two terms

$$\begin{aligned}
\mathbb{E}\left[\alpha^2(Z)\{\hat{\mu}(Z) - \mu(Z)\}^2\right] &< M||\hat{\mu} - \mu||^2 \\
\mathbb{E}\left[\{\hat{\alpha}(Z) - \alpha(Z)\}^2 \text{var}(Y|Z)\right] &< M||\hat{\alpha} - \alpha||^2
\end{aligned}$$

hence, these are also $o_p(1)$ by consistency.

Remark: The requirement for estimator independence can be relaxed if one makes Donsker class assumptions instead.

B.3. Proof of Theorem 1

Consider (1) with $\hat{\psi} = h_n(\hat{\mu}^*)$ and $\hat{\varphi}(W) = m(\hat{\mu}^*, W) + k\{\beta(Z) - \hat{\beta}_\perp(Z)\}\{Y - \hat{\mu}^*(Z)\}$, where we use the shorthand

$$k = \frac{h(\beta)}{||\beta - \beta_\perp||^2}$$

so that, by (5), $\alpha(Z) = k\{\beta(Z) - \beta_\perp(Z)\}$ and $\varphi(W) = m(\mu, W) + k\{\beta(Z) - \beta_\perp(Z)\}\{Y - \mu(Z)\}$. Under this parameterization, the plug-in bias on the right hand side of (1) is

$$\sqrt{n}\mathbb{E}_n[\hat{\varphi}(W) - \hat{\psi}] = \sqrt{n}k\mathbb{E}_n[\{\beta(Z) - \hat{\beta}_\perp(Z)\}\{Y - \hat{\mu}^*(Z)\}] = 0$$

Applying the result in Supplement B.1, the first-order remainder on the right hand side of (1) is

$$\sqrt{n}\mathbb{E}[\hat{\varphi}(W) - \Psi] = \sqrt{n}k\langle\hat{\mu}^* - \mu, \hat{\beta}_\perp - \beta_\perp\rangle = o_p(1)$$

Finally the second-order remainder on the right hand side of (1) is

$$\begin{aligned} G_n\left[\hat{\varphi}(W) - \varphi(W)\right] = G_n[m(\hat{\mu}^* - \mu, W)] \\ + kG_n\left[\{\hat{\beta}_\perp(Z) - \beta_\perp(Z)\}\{\hat{\mu}^*(Z) - \mu(Z)\}\right] \\ - kG_n\left[\{\beta(Z) - \beta_\perp(Z)\}\{\hat{\mu}^*(Z) - \mu(Z)\}\right] \\ - kG_n\left[\{\hat{\beta}_\perp(Z) - \beta_\perp(Z)\}\{Y - \mu(Z)\}\right] \end{aligned}$$

which is $o_p(1)$.

We have shown that plug-in bias, first-order remainder and second-order remainder in (1) are each $o_p(1)$, hence $h_n(\hat{\mu}^*)$ is RAL.

B.4. Sufficiency of learning conditional on the unscaled RR

Claim: $\Psi = h(\eta)$, where

$$\eta(z) \equiv \mathbb{E}[Y|\beta(Z) - \beta_\perp(Z) = \beta(z) - \beta_\perp(z)].$$

Proof:

$$\begin{aligned} \Psi &= \mathbb{E}[Y\alpha(Z)] \\ &= \frac{h(\beta)\mathbb{E}[Y\{\beta(Z) - \beta_\perp(Z)\}]}{||\beta - \beta_\perp||} \\ &= \frac{h(\beta)\mathbb{E}[\eta(Z)\{\beta(Z) - \beta_\perp(Z)\}]}{||\beta - \beta_\perp||} \\ &= \mathbb{E}[\eta(Z)\alpha(Z)] \end{aligned}$$

where in the third step we apply the law of iterated expectation.

B.5. Proof of orthogonality representation

Claim: Letting $\mu_\perp = \arg\min_{f \in \mathcal{C}^\perp} ||\mu - f||$

$$\mu(z) = \mu_\perp(z) + \frac{\Psi}{h(\beta)}\{\beta(z) - \beta_\perp(z)\}.$$

Proof: Note that $\mathcal{C}^\perp = \{f \in \mathcal{H} \mid \langle f, \alpha \rangle = 0\}$ then by Hilbert's projection theorem, $\mu_\perp$ exists, with

$$\mu_\perp(z) \equiv \mu(z) - \frac{\langle \mu, \alpha \rangle}{||\alpha||^2}\alpha(z) \quad \iff \quad \mu(z) = \mu_\perp(z) + \frac{\Psi}{h(\alpha)}\alpha(z)$$

Where we use $\langle \mu, \alpha \rangle = \Psi$ and $||\alpha||^2 = h(\alpha)$. Applying (5) completes the proof.