# A. Appendix

## A.1. Proof of Theorem 1

The next theorem formalizes this intuition and shows, in particular, that MODE-IV asymptotically identifies and consistently estimates the causal effect.

**Theorem 1.** *Fix a test point $(t, x)$ and let $\hat{\beta}_1, \ldots, \hat{\beta}_k$ be estimators of the causal effect of $t$ at $x$ corresponding to $k$ (possibly invalid) instruments. E.g., $\hat{\beta}_j = \hat{f}_j(t, x)$. Denote the true effect as $\beta = E[y|do(t), x]$. Suppose that*

1. *(consistent estimators) $\hat{\beta}_j \to \beta_j$ almost surely for each instrument. In particular, $\beta_j = \beta$ whenever the $j$th instrument is valid.*
2. *(modal validity) At least $v$ of the instruments are valid, and no more than $v - 1$ of the invalid instruments agree on an effect. That is, $v$ of the instruments yield the same estimand if and only if all of those instruments are valid.*

*Let $[\hat{l}, \hat{u}]$ be the smallest interval containing $v$ of the instruments and let $\hat{\mathcal{I}}_{\text{mode}} = \{i : \hat{l} \leq \hat{\beta}_i \leq \hat{u}\}$. Then,*

$$\sum_{i \in \hat{\mathcal{I}}_{\text{mode}}} \hat{w}_i \hat{\beta}_i \to \beta$$

*almost surely, where $\hat{w}_i, w_i$ are any non-negative set of weights such that each $\hat{w}_i \to w_i$ a.s. and $\sum_{i \in \hat{\mathcal{I}}_{\text{mode}}} w_i = 1$. Further if the individual estimators are also asymptotically jointly normal,*

$$\sqrt{n}[\hat{\beta}_1, \ldots, \hat{\beta}_k]^T \to N([\beta_1, \ldots, \beta_k]^T, \Sigma)$$

*with some covariance matrix $\Sigma \in \mathbb{R}^{k \times k}$. Then it also holds that the modal estimator is asymptotically normal:*

$$\sqrt{n} \sum_{i \in \hat{\mathcal{I}}_{\text{mode}}} \hat{w}_i \hat{\beta}_i \to N(\beta, w^T \Sigma_{\mathcal{I}_{mode}} w).$$

*where $\Sigma_{\mathcal{I}_{mode}} \in \mathbb{R}^{V \times V}$ denotes the covariance of the selected instruments.*

*Proof.* First we argue that $\hat{\mathcal{I}}_{\text{mode}}$ converges to a set that contains only valid instruments. All valid instruments converge to a common value $\beta$. The distance between any two valid instruments is at most twice the distance between $\beta$ and the furthest valid instrument. Since at least $v$ of the instruments are valid, this means that there is an interval (containing the mode) with distance going to 0 that contains $v$ of the instruments. Eventually this must be the smallest interval containing $v$ of the instruments, because the limiting $\beta_j$ of the invalid instruments are spaced out by assumption.

The result follows by continuous mapping. □

## A.2. Worst case rates

**Theorem 2.** *For some test point $(t, x)$, let $\hat{\beta}_1, \ldots, \hat{\beta}_k$ be $k$ estimates of the causal effect of $t$ at $x$. Assume,*

***[Bounded estimates]** Each estimate is bounded by some constants, $[a_i, b_i]$*

***[Convergent estimators]** Each estimator converges in mean squared error at a rate $n^{-r}$ (where $r = \frac{1}{2}$ if the estimator achieves the parametric rate), and hence each estimator has finite variance, $Var(\hat{\beta}_i) = \frac{\sigma_i}{n^{2r}}$ for some $\sigma_i$.*

*Then, there exists some constant, $C$, such that $E[(\text{ModeIV}(\mathcal{Z}) - \beta)^2] - E[(\frac{1}{v}\sum_{i \in \mathcal{V}} \hat{\beta}_i - \beta)^2] \leq 9kC\frac{\sigma}{n^r}$.*

*Proof.* Fix some test point $(t, x)$ and without loss of generality, assume that the true effect, $E[y|\text{do}(t), x] = 0$. Because ModeIV optimizes for the closest $V$ points, we can upper bound its worst case performance with an adversarial distribution that deterministically places all invalid instruments on the same point $a$ such that $\hat{\beta}_i = a$ for all $i$ in $\mathcal{I}$; assume $a > 0$ (again wlog, the argument is symmetric).

Consider the case where there are $v$ valid instruments and $v - 1$ invalid candidates. Let $d(\mathcal{V}) = \max_{i,j \in \mathcal{V}} |\hat{\beta}_i - \hat{\beta}_j|$ denote the largest distance between the any two estimates in the set of valid instruments, $\mathcal{V}$. Recall that, by definition, $\hat{\mathcal{I}}_{\text{mode}}$

is the set of the $v$ closest estimates, and let $d^* = \max_{i,j \in \hat{\mathcal{I}}_{\text{mode}}} |\hat{\beta}_i - \hat{\beta}_j|$ denote the largest distance among this set of $v$ estimates (regardless of whether or not they are valid). Now, if $\hat{\mathcal{I}}_{\text{mode}}$ includes the invalid candidates whose estimates are located at $a$, then $\hat{\mathcal{I}}_{\text{mode}}$ consists of $v-1$ invalid estimates and one valid estimate, $i$, for some $i \in \mathcal{V}$. In this case, $d^* = \min_{i \in \mathcal{V}} |\hat{\beta}_i - a| < d(\mathcal{V})$, and ModeIV returns $\frac{v-1}{v}a + \frac{1}{v}\hat{\beta}_i$; otherwise $d(\mathcal{V}) = d^*$ and ModeIV matches the oracle performance.

Now, consider the interval $C(a) = [-\frac{a}{3}, \frac{a}{3}]$; if all the valid instruments fall within $C(a)$ then we know that $d(\mathcal{V}) \leq \frac{2a}{3}$ and that the distance between $a$ and the closest valid estimate, $\min_{i \in \mathcal{V}} |\hat{\beta}_i - a| > \frac{2a}{3}$, so $d(\mathcal{V}) = d^*$ and ModeIV matches the oracle performance. So the difference between the oracle and ModeIV occurs when some valid estimates fall outside of $C(a)$. We bound this difference as follows,

$$\begin{aligned} E\left[(\text{ModeIV}(\mathcal{Z}))^2\right] - E\left[\left(\frac{1}{v}\sum_{i \in \mathcal{V}} \hat{\beta}_i\right)^2\right] &\leq \Big(1 - P(\hat{\beta}_i \in C(a) \text{ for all } i \in \mathcal{V})\Big)\left(\frac{v-1}{v}a + \frac{1}{v}\hat{\beta}_i\right)^2 \\ &\leq \Big(1 - P(\hat{\beta}_i \in C(a) \text{ for all } i \in \mathcal{V})\Big)\, a^2 \end{aligned} \tag{1}$$

Since each $\hat{\beta}_i$ is bounded $\in [a_i, b_i]$, we know they are $\tilde{\sigma}_i$-sub-Gaussian random variables for some $\tilde{\sigma}_i \leq \frac{a_i - b_i}{4}$. Let $\tilde{\sigma} = \max_{i \in \mathcal{V}} \tilde{\sigma}_i$ and note that there exists some constant $C$, such that for all $n$ and $i$, $\tilde{\sigma} \leq C\sigma_i n^{-r}$. Now,

$$\begin{aligned} P(\hat{\beta}_i \in C(a) \text{ for all } i \in \mathcal{V}) &= \prod_{i=1}^{k} P\left(\hat{\beta}_i \in C(a)\right) = \prod_{i=1}^{k}\left(1 - P\left(|\hat{\beta}_i| > \frac{a}{3}\right)\right) \geq \prod_{i=1}^{k}\left(1 - \exp\left(-\frac{a^2}{18\tilde{\sigma}_i}\right)\right) \\ &\geq \left(1 - \exp\left(-\frac{a^2}{18\tilde{\sigma}}\right)\right)^k \\ &\geq 1 - k\exp\left(-\frac{a^2}{18\tilde{\sigma}}\right) \end{aligned}$$

where the first inequality applies a tail bound on sub-Gaussian random variables and last inequality uses the fact that $(1+x)^k \geq 1 + kx$ for $x > -1$ and $k \geq 1$.

By substituting back into equation (1), we see that,

$$E\left[(\text{ModeIV}(\mathcal{Z}))^2\right] - E\left[\left(\frac{1}{v}\sum_{i \in \mathcal{V}} \hat{\beta}_i\right)^2\right] \leq a^2 k \exp\left(-\frac{a^2}{18\tilde{\sigma}}\right)$$

so we can get worst case performance by optimizing over, $a$. Taking partial deviates with respect to $a$ and setting it to zero, we get,

$$\left[2y^* k - \frac{k(a^*)^3}{9\tilde{\sigma}}\right]\exp\left(-\frac{(a^*)^2}{18\tilde{\sigma}}\right) = 0 \quad \rightarrow \quad a^* = 3\sqrt{2\tilde{\sigma}}$$

is the only local maximum such that $a > 0$. And hence, the worst case mean square error is bounded by,

$$E[(\text{ModeIV}(\mathcal{Z}))^2] - E\left[\left(\frac{1}{v}\sum_{i \in \mathcal{V}} \hat{\beta}_i\right)^2\right] \leq \frac{18}{e}k\tilde{\sigma} \leq 9k\tilde{\sigma} \leq 9kC\frac{\sigma}{n^r}$$

□

| | 50 / 100 valid | 60 / 100 valid | 70 / 100 valid | 80 / 100 valid | 90 / 100 valid | 100 / 100 valid |
|---|---|---|---|---|---|---|
| DeepIV (opt) | 0.036 ± (0.0048) | 0.042 ± (0.003) | 0.0371 ± (0.0032) | 0.0364 ± (0.003) | 0.0326 ± (0.0021) | 0.0278 ± (0.0021) |
| MODE-IV 20 | 0.0525 ± (0.0062) | 0.0483 ± (0.0049) | 0.0478 ± (0.0049) | 0.0473 ± (0.0048) | 0.0466 ± (0.0047) | 0.04 ± (0.0038) |
| MODE-IV 30 | 0.0525 ± (0.0062) | 0.0483 ± (0.0049) | 0.0478 ± (0.0049) | 0.0473 ± (0.0048) | 0.0467 ± (0.0047) | 0.0399 ± (0.0038) |
| MODE-IV 40 | 0.0524 ± (0.0062) | 0.0483 ± (0.0049) | 0.0478 ± (0.0049) | 0.0473 ± (0.0048) | 0.0468 ± (0.0047) | 0.0399 ± (0.0038) |
| MODE-IV 50 | 0.0525 ± (0.0062) | 0.0483 ± (0.0049) | 0.0479 ± (0.0049) | 0.0474 ± (0.0048) | 0.0466 ± (0.0047) | 0.0398 ± (0.0038) |
| Mean | 0.0529 ± (0.0059) | 0.0498 ± (0.005) | 0.0498 ± (0.0052) | 0.0461 ± (0.0047) | 0.0484 ± (0.0048) | 0.0403 ± (0.004) |
| Deepiv (all) | 0.1637 ± (0.011) | 0.1744 ± (0.0075) | 0.2078 ± (0.0069) | 0.185 ± (0.0087) | 0.1387 ± (0.0081) | 0.0297 ± (0.0018) |

*Table 3.* Average absolute bias in estimation of the conditional average treatment effect. The ensemble methods tended to have slightly larger bias than the optimal model, but far less than the naive approach which uses all instruments. The mean aggregation function performs relatively well on this task, but this approach comes with no guarantees, so it degrades in settings with more bias.

### A.3. Additional experimental details

**Network architectures and experimental setup** All experiments used the same neural network architectures to build up hidden representations for both the treatment and response networks used in DeepIV, and differed only in their final layers. Given the number of experiments that needed to be run, hyper-parameter tuning would have been too expensive, so the hyper-parameters were simply those used in the original DeepIV paper. In particular, we used three hidden layers with 128, 64, and 32 units respectively and ReLU activation functions. In the Mendelian randomization experiments, we used The treatment networks all used mixture density networks with 10 mixture components and the response networks were trained using the two sample unbiased gradient loss (see Hartford et al., 2017, equation 10). We used our own PyTorch (Paszke et al., 2019) re-implementation of DeepIV to run the experiments.

For all experiments, $10\%$ of the original training set was kept aside as a validation set. Both the demand and Mendelian randomization simulations had 90 000 training examples, 10 000 validation examples and 50 000 test examples. All mean squared error numbers reported in the paper are calculated with respect to the true $y$ (with no confounding noise added) on a uniform grid of 50 000 treatment points between the $2.5^{th}$ and $97.5^{th}$ percentiles of the training distribution. As a result, we are measuring mean squared error relative to the target $y$ on the uniform grid of treatment values corresponds to $E[y|\text{do}(t'), x, z]$ for each $t'$ on the grid. 95% confidence intervals around mean performance were computed using Student's $t$ distribution.

The networks were trained on a large shared compute cluster that has around 100 000 CPU cores. Because each individual network was relatively quick to train (less than 10 minutes on a CPU), we used CPUs to train the networks. This allowed us to fit the large number of networks needed for the experiments. Each experiment was run across 30 different random seeds, each of which required 10 (demand simulation) and 102 (Mendialian randomization) network fits. In total, across all experimental setups, random seeds, ensembles, etc. approximately 100 000 networks were fit to run all the experiments.

**Biased demand simulation** The biased demand simulation code was modified from the public DeepIV implementation. In the original DeepIV implementation, both the treatment and response are transformed to make them approximately mean zero and standard deviation 1; we left these constants unchanged ($t_{\text{std}} = 3.7, t_\mu = 17.779, y_{\text{std}} = 158, y_\mu = -292.1$). The observed features include a time feature, $x_0 \sim \text{unif}(0, 10)$, and $x \sim \text{Categorical}(\frac{1}{7}, \ldots, \frac{1}{7})$, a one-hot encoding of 7 different equally likely "customer types"; each type modifies the treatment via the coefficient $\beta^{(x)} = [1, 2, \ldots, 7]^T$. These values are unchanged from the original Hartford et al. data generating process. We introduce multiple instruments, $z_{1:k}$, whose effect on the treatment, $t$, and response, $y$, is via two different linear maps $\beta^{(zt)} \in \Re^k$ and $\beta^{(zy)} \in \Re^k$; each of the coefficients in these vectors are sampled independently so $\beta_i^{(z*)} \sim \text{unif}(0.5, 1.5)$, with the exception of the valid instruments where $\beta_i^{(zy)} = 0$ for all $i \in \mathcal{V}$. The $\gamma$ parameter scales the amount of bias introduced via exclusion violations; note that because the $\sin$ varies between $[-1, 1]$, we scale up this bias by a factor of 60 so that the effect it introduces is of the same order of magnitude as the variation in the original Hartford et al. data generating process (where std. dev$(y) \approx y_{\text{std}} = 158$).

The full data generating process is as follows,

$$z_{1:k}, \nu \sim \mathcal{N}(0,1) \quad x_0 \sim \text{unif}(0,10) \quad e \sim \mathcal{N}(\rho\nu, 1-\rho^2), \quad x \sim \text{Categorical}(\frac{1}{7},\ldots,\frac{1}{7})$$
$$t' = 25 + (z^T\beta^{(zt)} + 3)\psi(x_0) + \nu$$
$$y' = 100 + 10x^T\beta^{(x)}\psi(x_0) + \underbrace{(x^T\beta^{(x)}\psi(x_0) - 2)}_{\text{Treatment effect}}t' + \underbrace{\gamma 60\sin(z^T\beta^{(zy)})}_{\text{Exclusion violation}} + e$$
$$t = (t' - t_{\text{std}})/t_\mu \qquad y = (y' - y_{\text{std}})/y_\mu$$

**Mendelian randomization simulation** This data generating process closely follows that of Simulation 1 of Hartwig et al. (2017), but was modified to include heterogeneous treatment effects. This description is an abridged version of that given in Hartwig et al.; we refer the reader to Hartwig et al. (2017) for more detail on the choice of parameters, etc.. The instruments are the genetic variables, $z_i$, which were generated by sampling from a Binomial (2, $p_i$) distribution, with $p_i$ drawn from a Uniform(0.1 ,0.9) distribution, to mimic bi-allelic SNPs in Hardy-Weinberg equilibrium. The parameters that modulate the genetic variable effect on the treatment are given by, $\alpha_i = \frac{\sqrt{0.1}}{\sigma_{zx}}\nu_i$, where $\nu_i \sim \text{unif}(0.01, 0.2)$ and $\sigma_{zx} =$ std. dev$(\sqrt{0.1}\sum_i \nu_i z_i)$. Similarly, the exclusion violation parameters, $\delta_i = \frac{|\mathcal{I}|\sqrt{0.1}}{k\sigma_{zy}}\nu_i$, where again $\nu_i \sim \text{unif}(0.01, 0.2)$ and $\sigma_{zy} =$ std. dev$(\sqrt{0.1}\sum_i \nu_i z_i)$. Note that $\delta_i$ is scaled by $\frac{|\mathcal{I}|}{k}$ (the proportion of invalid instruments), which ensures that the average amount of bias introduced is constant as the number of invalid instruments vary. Error terms $u$, $\epsilon_x$, $\epsilon_y$ were independently generated from a normal distribution, with mean 0 and variances $\sigma_u^2, \sigma_y^2$ and $\sigma_y^2$, respectively, whose values were chosen to set the variances of $u$, $x$ and $y$ to one. These scaling parameters are chosen to enable an easy interpretation of the average treatment effect, $\beta$: with this scaling, $\beta = 0.1$ implies that one standard deviation of $t$ causes a $0.1$ standard deviation of $y$ and hence the causal effect of $t$ on $y$ explains $0.1^2 = 0.01 = 1\%$ of the variance of $y$. The only place our simulation differs from Hartwig et al., is the treatment effect is a function of observed coefficients, $x \in \Re^{10}$, with each $x_i \sim$ uniform$(-0.5, 0.5)$, and the treatment effect is defined as, $\beta(x) :=$ round$(x^T\gamma^{(xt)}, 0.1)$, with $\gamma^{(xt)}$ a sparse vector of length 10, with three non-zeros $\gamma_i^{(xt)} \sim$ uniform$(0.2, 0.5)$. The resulting true $\beta(x)$, takes on values in $\{-0.3, -0.2, \ldots, 0.2, 0.3\}$. We also use 100 genetic variants instead of the 30 used in Hartwig et al., so that we could test ModeIV in a larger scale scenario. The resulting data generating process is given by,

$$z_i \sim \text{Binomial}(2, p_i) \quad \text{for } i \text{ in } [1\ldots K], \quad \beta(x) := \text{round}(x^T\gamma^{(xt)}, 0.1).$$
$$t := \sum_{j=1}^{K}\alpha_j z_j + \rho u + \epsilon_x$$
$$y := \beta(x)t + \sum_{j=1}^{K}\delta_j z_j + u + \epsilon_y$$

### A.4. Details on the bootstrap experiments

For all the bootstrap experiments we estimated a bootstrap confidence interval point-wise so, at a given point $(t, x, z)$ the interval is computed as,

$$CI_\alpha(t,x,z) = \bar{\hat{f}}(t,x,z) \pm \text{z}_{\alpha/2}\text{se}(\hat{f}(t,x,z)) \text{ where } \bar{\hat{f}}(t,x,z) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}(t,x,z;D_b^*),$$
$$\text{se}(\hat{f}(t,x,z)) = \sqrt{\frac{1}{B}\sum_{b=1}^{B}\left(\hat{f}(t,x,z;D_b^*) - \bar{\hat{f}}(t,x,z)\right)^2}$$

We used $B = 50$ bootstrap samples where for each bootstrap sample of the dataset $D_b^*$, we run the full ModeIV procedure (i.e. we fit $k$ estimates of DeepIV on $D_b^*$, each with a different one of the the $k$ instruments). $\text{z}_{\alpha/2}$ denotes the $\alpha/2$ quantile of a standard normal distribution.

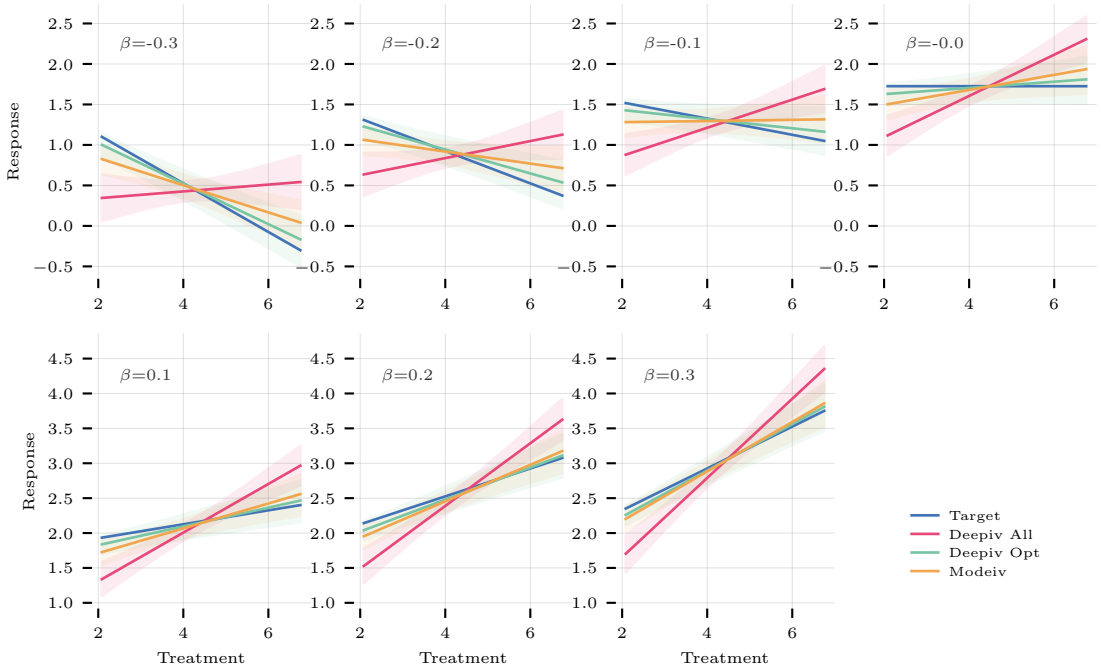


*Figure 7.* Bootstrap 90% confidence intervals of conditional dose-response curves. These plots show the average prediction and intervals, averaged across values of the conditioning variable, $x$, such that the true slope, $\beta(x)$, is given by the value indicated in the plots.

We compute coverage by evaluating the proportion of samples for which the true $y$ falls within an estimated bootstrap confidence interval,

$$E[\mathbf{1}(y \in CI_\alpha(t, x, z))] \approx \frac{1}{n} \sum_i \mathbf{1}(y_i \in I_\alpha(t_i, x_i, z_i))$$

For both the demand and Mendelian randomization experiments we used $n = 10\,000$ test points.

**Demand simulations** The bootstrap results presented in Table 1 give empirical coverage estimates for the demand simulation with between 4/7 and 6/7 valid instruments. We repeat the experiment across 20 random seeds and report average coverage across those experiments.

**Mendelian randomization** The Mendelian randomization experiments were too computationally expensive to give bootstrap inference results across the whole range of valid instruments and random seeds that we tested in point estimate experiments. Instead, we just tested coverage with 70% valid instruments on a single random seed[7]. The results are summarized in Table 4 and Figure 7.

Figure 7 shows the confidence intervals for the conditional does-response curves. Recall that in this simulation, the conditional average treatment effect is given by $\beta(x) := \text{round}(x^T \gamma^{(xt)}, 0.1)$, where $\gamma^{(xt)}$ are some unknown parameters, and the rounding ensures that $\beta(x)$ can only take on a fixed number of discrete values. This approach makes visualizing conditional average treatment effects easier as we can examine the average predicted effect for each of the different ground truth values of $\beta(x)$. Of course, these plots are only possible to make given the knowledge of the ground truth $\beta(x)$, so they are only useful as a diagnostic tool in simulated data, not something that we would be able to plot on real data. These conditional does-response curves are given by,

$$E[\hat{y}|t, \beta'] = E_z[E_{x:\beta(x)=\beta'}[\hat{f}(t, x, z)|z]|t, \beta]$$

That is, we average the predictions over all $x$ such that the true value of $\beta(x)$ is $\beta'$. Note that the curves are linear because we parameterize the deep network such the the condition treatment-response relationship is linear. The confidence intervals

[7] i.e. the random seed for the data generating process was fixed to 1 (chosen arbitrarily) which fixes the $\gamma$, $\delta$ and $\alpha$ parameters

| Model | Coverage (70 / 100 valid) |
|---|---|
| ModeIV-10 | 64.76% |
| ModeIV-20 | 63.94% |
| ModeIV-30 | 62.64% |
| ModeIV-40 | 60.38% |
| ModeIV-50 | 59.78% |
| DeepIV-Opt | 80.96% |
| DeepIV-All | 30.96% |

*Table 4.* Mendelian randomization empirical coverage for 90% bootstrap confidence intervals. All the approaches underestimate coverage: the true $y$ falls with the interval 80% of the time, while the various Mode-IV approaches get between 60% and 65%. This is far better than DeepIV-all which only manages 30% because of its far more significant bias.

are computed in the same manner,

$$E[CI_\alpha(\hat{y})|t, \beta'] = E_z[E_{x:\beta(x)=\beta'}[CI_\alpha(t, x, z)|z]|t, \beta]$$

The plots show that both ModeIV and DeepIV-Opt do a good job of recovering the ground-truth relationship, particularly for positive values of the ground truth relationship. For negative values, some bias remains, which is the likely cause of the poor coverage numbers shown in Table 4. The fact that even DeepIV-opt underestimates coverage, suggests this may be the result of finite-sample bias rather than a problem with ModeIV itself.

### A.5. Details on Two-Stage Hard Thresholding experiments

| # valid / 100 | ATE: $\hat{\beta}$ | True Positive | False Positive | True Negative | False Negative |
|---|---|---|---|---|---|
| 50 | 0.08 | 0.47 | 0.19 | 0.31 | 0.03 |
| 60 | 0.03 | 0.56 | 0.09 | 0.31 | 0.04 |
| 70 | 0.02 | 0.66 | 0.04 | 0.26 | 0.04 |
| 80 | 0.01 | 0.75 | 0.01 | 0.19 | 0.05 |
| 90 | 0.00 | 0.84 | 0.00 | 0.10 | 0.06 |
| 100 | 0.00 | 0.93 | 0.00 | 0.00 | 0.07 |

*Table 5.* Average treatement effect and confusion matrices for Guo et al. (2018)'s two-stage hard thresholding algorithm on the Mendelian randomization dataset. When 70 or more instruments were valid, it had a low false positive rate. As a result, it accurately recovered the average treatment effect for this simulation ($\beta = 0.$).

We used Guo et al. (2018)'s public R implementation of Two-Stage Hard Thresholding (available here). To estimate $E[y|\text{do}(t), x]$, we ran two-stage least squares on the predicted valid instruments while controlling for the invalid candidates. All experiments were run 30 times with different seeds for the data generating process and we report mean performance.

**ModeIV for linear constant treatment effects**

We also evaluated ModeIV on Guo et al. (2018)'s low-dimensional data generating process.[8] We used the same network architecture as the MR experiments with a conditionally linear output so we could observe the coefficient on $t$ (i.e. $\hat{\beta}$) directly. Table 6 reports percentiles of the estimated ATE with $n$ of 1000 and 10000. This is a massively over-parameterized model for the task (roughly 20 000 parameters for each of the 10 candidates; note that you should never do this if you know the problem is linear!). As expected, ModeIV was less efficient but with 10 000 training examples it picked up only a small amount of bias, $\hat{\text{ATE}} = 1.039$ (for a true ATE of 1). With 1000 samples ModeIV incurred large bias from the invalid candidates.

### A.6. Dalenius–Venter mode implementation

[8]Note -Guo et al.'s high-dimensional DGP involves a large number of candidates that fail the relevance assumption. Given that relevance is testable and ModeIV is computationally expensive, relevance testing is better addressed as a pre-processing step.

| Model | $n$ | $\hat{\beta}_{10}$ | $\hat{\beta}_{25}$ | $\hat{\beta}_{50}$ | $\hat{\beta}_{75}$ | $\hat{\beta}_{90}$ |
|---|---|---|---|---|---|---|
| ModeIV-0.2 | 1000 | 1.043 | 1.151 | 1.285 | 1.471 | 2.565 |
| | 10000 | 0.989 | 1.013 | 1.044 | 1.087 | 1.249 |
| ModeIV-0.3 | 1000 | 1.056 | 1.151 | 1.270 | 1.423 | 1.734 |
| | 10000 | 0.993 | 1.013 | 1.040 | 1.075 | 1.132 |
| ModeIV-0.4 | 1000 | 1.066 | 1.155 | 1.267 | 1.412 | 1.665 |
| | 10000 | 0.997 | 1.015 | 1.039 | 1.072 | 1.125 |
| ModeIV-0.5 | 1000 | 1.079 | 1.161 | 1.268 | 1.415 | 1.667 |
| | 10000 | 1.000 | 1.016 | 1.039 | 1.072 | 1.128 |
| DeepIV-opt | 1000 | 0.908 | 0.980 | 1.059 | 1.142 | 1.243 |
| | 10000 | 0.976 | 0.987 | 1.004 | 1.019 | 1.036 |

*Table 6.* With enough data, ModeIV performs will in the linear setting. Here we show performance on Guo et al. (2018)'s data generating process. The true $\beta = 1$; because ModeIV controls for invalid candidates, there is some variation in predicted $\hat{\beta}$ across examples; we show this with percentiles of the $\hat{\beta}$ estimates which are denoted $\hat{\beta}_p$.

```
def estimate_mode(predictions, v):
    '''
    Estimate the Dalenius-Venter mode.

    Args:
        predictions: A tensor of size (batch_size, ensemble_size)
                     containing the predictions from each ensemble member.
        v: The number of elements that will form part of the modal interval.
           2 <= v <= ensemble_size.

    Returns:
        The Dalenius / Venter mode - the mean of the v closest predictions.
    '''
    k = predictions.shape[1]
    sort_pred, _ = torch.sort(predictions, axis=1)
    min_idx = torch.argmin(sort_pred[:,v-1:] - sort_pred[:,:(k - v + 1)], axis=1)
    modal_indices = torch.cat([min_idx[:,None] + i for i in range(v)], dim=1)
    return torch.gather(sort_pred, 1, modal_indices).mean(axis=1)
```