Causal Inference with Unstructured Data

Germain Gauthier
(Bocconi University)

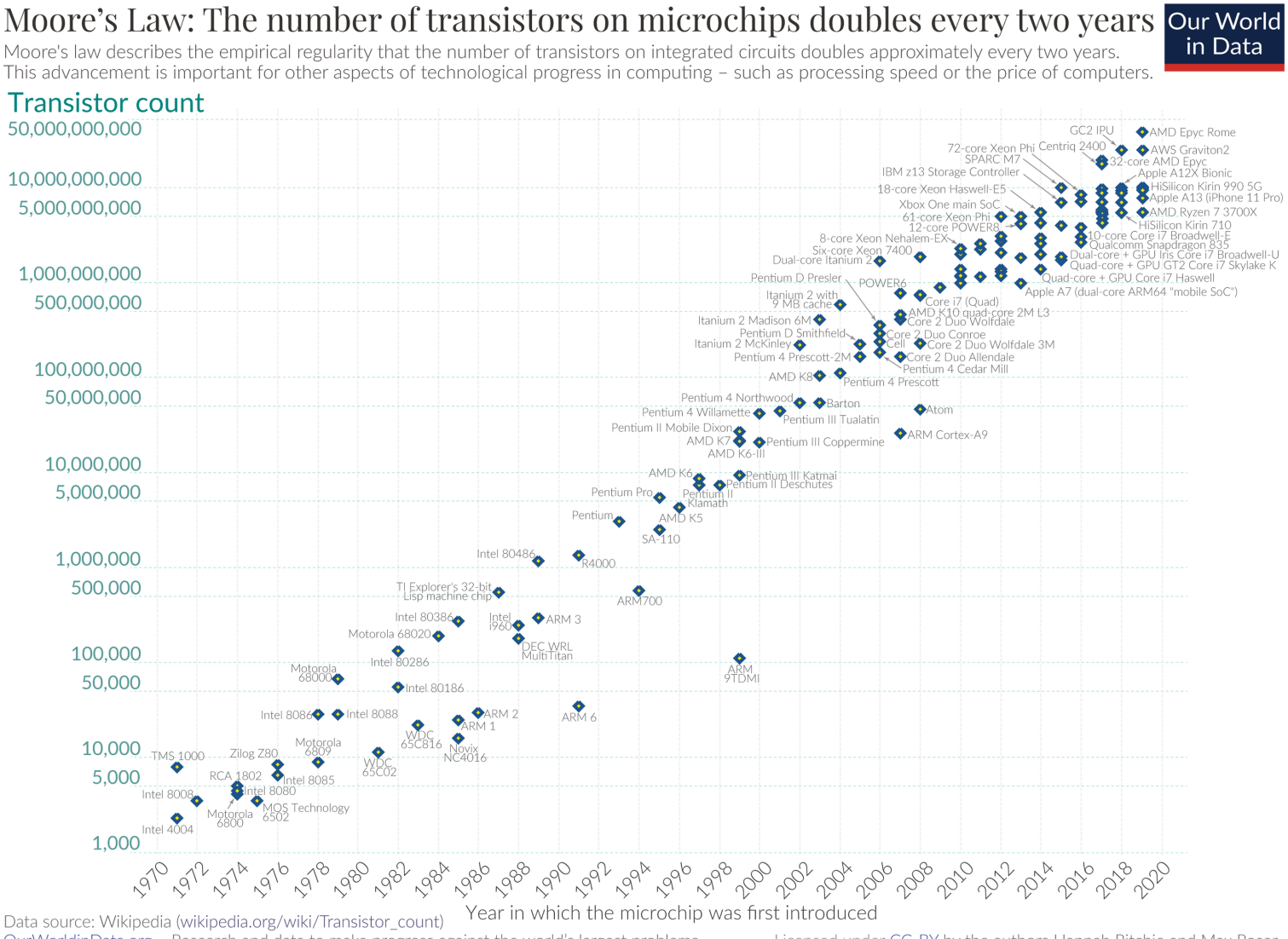
September 2026

Outline

Unstructured data is plentiful and compute is cheap.

- The digital era generates A LOT of unstructured data.
  - e.g., Social media posts, Internet queries, Wikipedia, online newspapers, TV transcripts, digitized books, political speeches and manifestos, laws, images, podcasts, videos, etc.
- It is matched with a similar increase in computational resources.
  - Moore's law = number of transistors on microchips doubles every two years (since the 70s!)
- Nowadays, many applied economics papers use ML/AI to measure concepts in unstructured data.
  - e.g., economic policy uncertainty, news sentiment, racial and misogynistic bias, political and economic narratives, speech polarization, etc.
  - Some useful literature reviews: Gentzkow et al. (2019); Ash and Hansen (2023); Dell (2025); Ludwig et al. (2026)

Moore’s Law



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

Very often, applied papers treat ML/AI-generated variables as data.

Step 1: Measurement

Turn unstructured data into a structured variable, $\widehat{Z}_i$.

Step 2: Inference

Plug $\widehat{Z}_i$ into a regression, DiD, IV, RDD, synthetic control, etc.

$$\underbrace{U_i}_{\substack{\text{Text, image,}\\ \text{audio, video}}} \xrightarrow{\text{AI/ML annotator } \widehat{g}} \underbrace{\widehat{Z}_i = \widehat{g}(U_i)}_{\text{Generated variable}} \xrightarrow{\text{Standard causal estimator}} \widehat{\theta}_{\text{naive}}$$

So researchers often treat $\widehat{Z}_i$ as the directly observed Z_i. There's a fair chance that is not a good idea...!

In this lecture, we review the fast-growing literature that focuses on the econometric issues this raises and offers some solutions.

Main references: Angelopoulos et al. (2023); Egami et al. (2023); Battaglia et al. (2025); Carlson and Dell (2026); Ludwig et al. (2026)

Some useful distinctions to get us started.

Below are a few questions I want you to keep in the back of your mind during this lecture:

- Is the AI/ML-generated variable an outcome or a regressor in the downstream regression?
- Do groundtruth labels of the AI/ML-generated variable exist?
  - Supervised learning setting
  - e.g., The groundtruth labels are observed for a share of the sample (usually human annotations).
- If not, what latent structure identifies it?
  - Unsupervised learning setting
  - The authors need to posit a structural model of the latent variable and its relationship to unstructured data.
  - e.g., a topic model, an ideal point model, etc.

Outline

A simple example

Imagine units are randomly assigned to a treatment, $D_i \in \{0, 1\}$. Each unit produces a document, U_i, whose expert label is Y_i^*. Random assignment identifies

$$\tau = \mathbb{E}[Y_i^* \mid D_i = 1] - \mathbb{E}[Y_i^* \mid D_i = 0].$$

The researcher labels every document with an AI/ML annotator (e.g., an ML classifier or an LLM):

$$\widehat{Y}_i = \widehat{g}(U_i) = Y_i^* + e_i, \qquad e_i = \widehat{Y}_i - Y_i^*.$$

The researcher also has expert labels for a random sample of documents drawn with probability π. She reports that $\widehat{g}$ "performs well" on this sample (e.g., a high correlation with the expert labels).

The researcher then treats $\widehat{Y}_i$ as observed truth and reports

$$\widehat{\tau}_{\text{naive}} = \overline{\widehat{Y}}_{D=1} - \overline{\widehat{Y}}_{D=0}.$$

When is there bias?

At the population level,

$$\begin{aligned}\tau_{\text{naive}} &= \mathbb{E}[\widehat{Y}_i \mid D_i = 1] - \mathbb{E}[\widehat{Y}_i \mid D_i = 0] \\ &= \tau + \underbrace{\mathbb{E}[e_i \mid D_i = 1] - \mathbb{E}[e_i \mid D_i = 0]}_{\text{bias}}.\end{aligned}$$

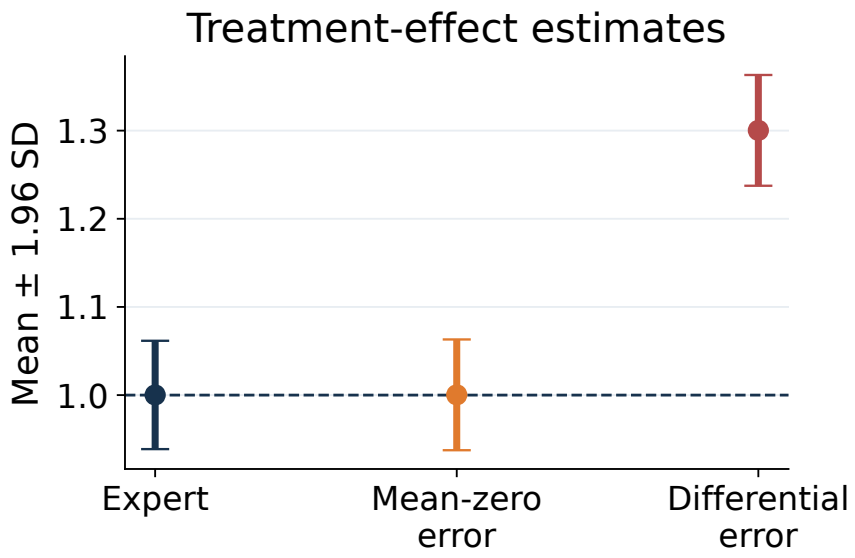
There is no bias if prediction errors have the same mean in both groups:

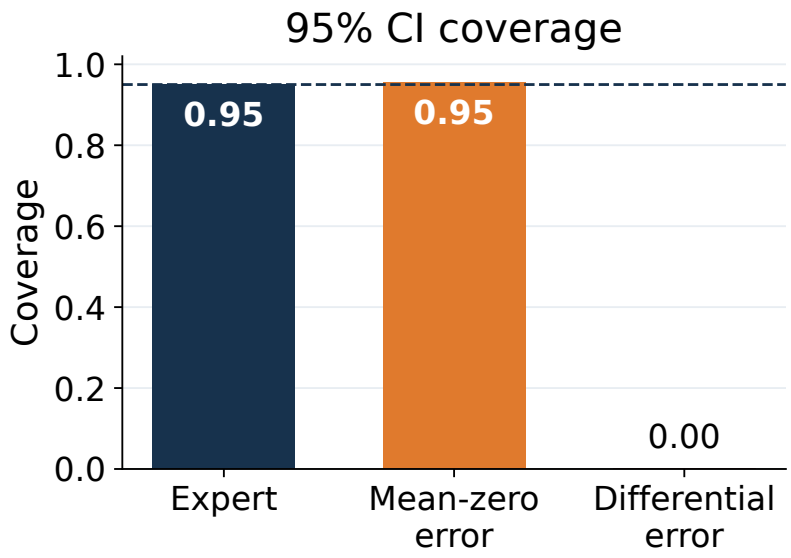
$$\mathbb{E}[e_i \mid D_i = 1] = \mathbb{E}[e_i \mid D_i = 0].$$

There is bias when prediction errors differ systematically by treatment. For example, if the model rates treated units' documents an average 0.30 points higher,

$$e_i = 0.30 D_i + \nu_i, \quad \mathbb{E}[\nu_i \mid D_i] = 0 \quad \Longrightarrow \quad \tau_{\text{naive}} = \tau + 0.30.$$

Simulation

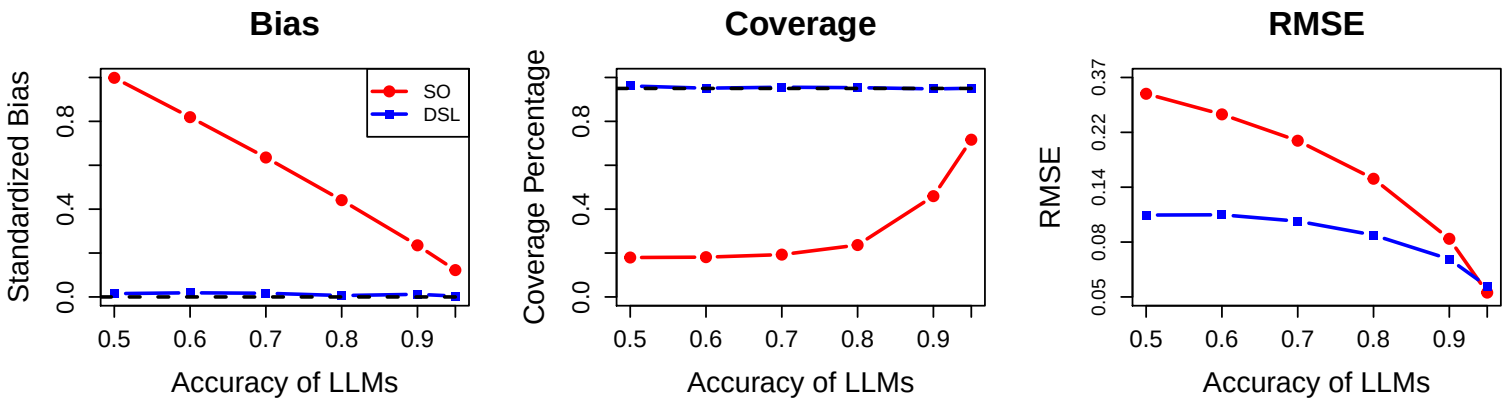




$n = 4{,}000$, 1,000 simulations, and $Y_i^* = D_i + \varepsilon_i$. The differential-error DGP uses $\widehat{Y}_i = Y_i^* + 0.30 D_i + \nu_i$.

High predictive accuracy is not enough.

Importantly, the bias can be substantial even when the AI label is highly correlated with the expert label:



In Egami et al. (2023), even at 90% classification accuracy, the surrogate-only 95% confidence interval covers the truth only about 40% of the time. Source: Figure 1a.

So... what should we do? Well, we have ground truth labels for some observations, so let's use them!

Setup and Notations

For every observation i we observe unstructured data U_i, covariates D_i (e.g., a treatment), and the AI/ML annotation $\widehat{Y}_i = \widehat{g}(U_i)$. Collect these observables in $V_i = (U_i, D_i, \widehat{Y}_i)$. For a subsample ($R_i = 1$) we also observe the ground-truth label Y_i^*.

A1 **Labels are the ground truth.** The target is defined by Y_i^* (e.g., an expert annotation): truth relative to a chosen measurement protocol.

A2 **The validation sample is representative.** In our running setup, ground-truth labels are collected on a simple random sample:

$$R_i \perp (Y_i^*, V_i), \qquad \Pr(R_i = 1) = \pi > 0.$$

What we do not assume: that the AI/ML annotator is accurate or unbiased. Arbitrary bias and errors correlated with treatment are allowed.

This baseline setup is sufficient for all four methods below; some allow more general validation sampling designs (I will let you know).

Core idea: use the subsample with ground-truth labels to correct for the measurement error in the AI/ML annotations.

The four papers implement this correction at different levels:

1. **Ludwig et al.** correct the *coefficient*: subtract the regression of the measurement error on the covariates.
2. **Design-based Supervised Learning (DSL)** corrects the *outcome*: an AIPW pseudo-outcome replaces the missing labels.
3. **Prediction-powered Inference (PPI)** corrects the *estimating score*: a rectifier fixes the moment condition.
4. **Missing at Random Structured Data (MAR-S)** corrects the *influence function* in a semiparametric missing-data framework.

A simple debiasing approach at the **coefficient** level (Ludwig et al., 2026)

A simple random subsample has the expert labels (π constant).

Let $W_i = (1, D_i)'$. The target is the coefficient vector β^* from regressing the ground-truth label Y_i^* on W_i. The three relevant population regressions are

$$\begin{aligned} Y_i^* &= W_i'\beta^* + \varepsilon_i \quad \text{(target regression)}, \\ \widehat{Y}_i &= W_i'\beta_{\widehat{Y}} + u_i \quad \text{(primary sample)}, \\ e_i \equiv \widehat{Y}_i - Y_i^* &= W_i'\beta_e + v_i \quad \text{(labeled subsample)}. \end{aligned}$$

Because $Y_i^* = \widehat{Y}_i - e_i$, linear projections imply

$$\beta^* = \beta_{\widehat{Y}} - \beta_e \quad \Longrightarrow \quad \widehat{\beta}_{\text{debiased}} = \widehat{\beta}_{\widehat{Y},\text{primary}} - \widehat{\beta}_{e,\text{labeled}}.$$

DSL directly works with **pseudo-outcomes** (Egami et al., 2023).

Using a cross-fitted imputation model $\widehat{m}(V_i)$ (details in the appendix), DSL constructs an augmented inverse-probability weighted (AIPW) pseudo-outcome:

$$\widetilde{Y}_i = \underbrace{\widehat{m}(V_i)}_{\substack{\text{predicted label} \\ \text{(often simply } \widehat{Y}_i\text{)}}} + \frac{\overbrace{R_i}^{\substack{\text{observation has} \\ \text{a ground-truth label}}}}{\underbrace{\pi(V_i)}_{\substack{\text{prob. of being} \\ \text{expert-labeled}}}} \underbrace{\{Y_i^* - \widehat{m}(V_i)\}}_{\substack{\text{measurement error:} \\ \text{ground truth vs. AI/ML}}}$$

then plugs $\widetilde{Y}_i$ into the target estimating equation; in the running example, this is just the regression of $\widetilde{Y}_i$ on W_i.

Imputation: a predicted label for every observation. **Correction:** for labeled observations, the observed prediction error $Y_i^* - \widehat{m}(V_i)$, reweighted by $1/\pi(V_i)$ to stand for the full sample ($\pi = 0.05$: each error counts 20 times). Sampling can be stratified or oversampled, as long as the researcher knows $\pi(V_i)$.

▸ Why it works ▸ Cross-fitting $m(V)$

PPI directly corrects the **estimating score** (Angelopoulos et al., 2023).

In the baseline PPI setup, the AI predictor $\widehat{g}$ is fixed (pre-trained). Suppose θ^* is identified by the estimating score

$$\mathbb{E}[s_i(Y_i^*; \theta^*)] = 0.$$

PPI adds and subtracts the score computed with the ML/AI prediction:

$$\mathbb{E}[s_i(Y_i^*; \theta)] = \underbrace{\mathbb{E}[s_i(\widehat{Y}_i; \theta)]}_{\substack{\text{Imputation:} \\ \text{AI labels, all observations}}} + \underbrace{\mathbb{E}\Big[s_i(Y_i^*; \theta) - s_i(\widehat{Y}_i; \theta)\Big]}_{\substack{\text{Correction: the average score error} \\ \text{induced by the AI labels,} \\ \text{estimated on the labeled sample}}}$$

Note that this equality holds for any AI/ML annotator that produces $\widehat{Y}_i$.

▸ Common estimands

MAR-S corrects the **influence function** (Carlson and Dell, 2026).

The estimand solves a moment condition, $\mathbb{E}[g(Z_i;\theta^*)] = 0$. Observation i's *influence function* is its rescaled moment,

$$\phi_i^{\text{full}}(\theta) \;:=\; -M^{-1}\, g(Z_i;\theta), \qquad M = \mathbb{E}[\partial_\theta\, g(Z_i;\theta^*)]\,.$$

Averaging the $\phi_i^{\text{full}}(\theta)$ measures how far θ is from θ^*, for any estimand:

$$\mathbb{E}\Big[\phi_i^{\text{full}}(\theta)\Big] \;\approx\; \theta^* - \theta \qquad \text{for } \theta \text{ near } \theta^*.$$

Missing labels leave holes in that average. Let $\widehat{m}_\phi(V_i;\theta) \approx \mathbb{E}\big[\phi_i^{\text{full}}(\theta) \mid V_i\big]$ denote a predictor of observation i's influence contribution:

$$\widehat{\phi}_i^{\text{MAR-S}}(\theta) = \underbrace{\widehat{m}_\phi(V_i;\theta)}_{\substack{\text{Imputation:}\\ \text{predicted error contribution,}\\ \text{all observations}}} + \frac{\overbrace{R_i}^{\substack{\text{has a ground-}\\ \text{truth label}}}}{\underbrace{\pi(V_i)}_{\substack{\text{prob. of being}\\ \text{labeled}}}} \underbrace{\Big\{\phi_i^{\text{full}}(\theta) - \widehat{m}_\phi(V_i;\theta)\Big\}}_{\substack{\text{Correction:}\\ \text{imputation error,}\\ \text{labeled sample}}}$$

Estimation in practice

With cross-fitted nuisances ($\widehat{\pi}$ and the imputation model $\widehat{\mathbb{E}}[\phi_i^{\text{full}} \mid V_i]$), start from any $\sqrt{n}$-consistent $\widetilde{\theta}$ and take one correction step:

$$\widehat{\theta} = \widetilde{\theta} + \frac{1}{n} \sum_{i=1}^{n} \widehat{\phi}_i^{\text{MAR-S}}(\widetilde{\theta}).$$

Why does the one-step work? Near θ^*,

$$\mathbb{E}[\phi_i^{\text{MAR-S}}(\theta)] \approx \theta^* - \theta.$$

The average $\widehat{\phi}_i^{\text{MAR-S}}(\widetilde{\theta})$ therefore estimates the error of $\widetilde{\theta}$; adding it removes that error to first order. Estimate your estimator's error, then subtract it: one Newton step is enough.

You can directly build standard errors from the variance of the $\widehat{\phi}_i^{\text{MAR-S}}(\widehat{\theta})$.

▸ Imputation term

MAR-S accommodates common empirical workflows.

Many downstream estimands

Carlson and Dell derive MAR-S estimators for:

- descriptive moments,
- linear regression,
- linear IV,
- difference-in-differences,
- regression discontinuity.

The generated variable can enter the analysis as an outcome, treatment, regressor, or control.

Aggregation and transformation

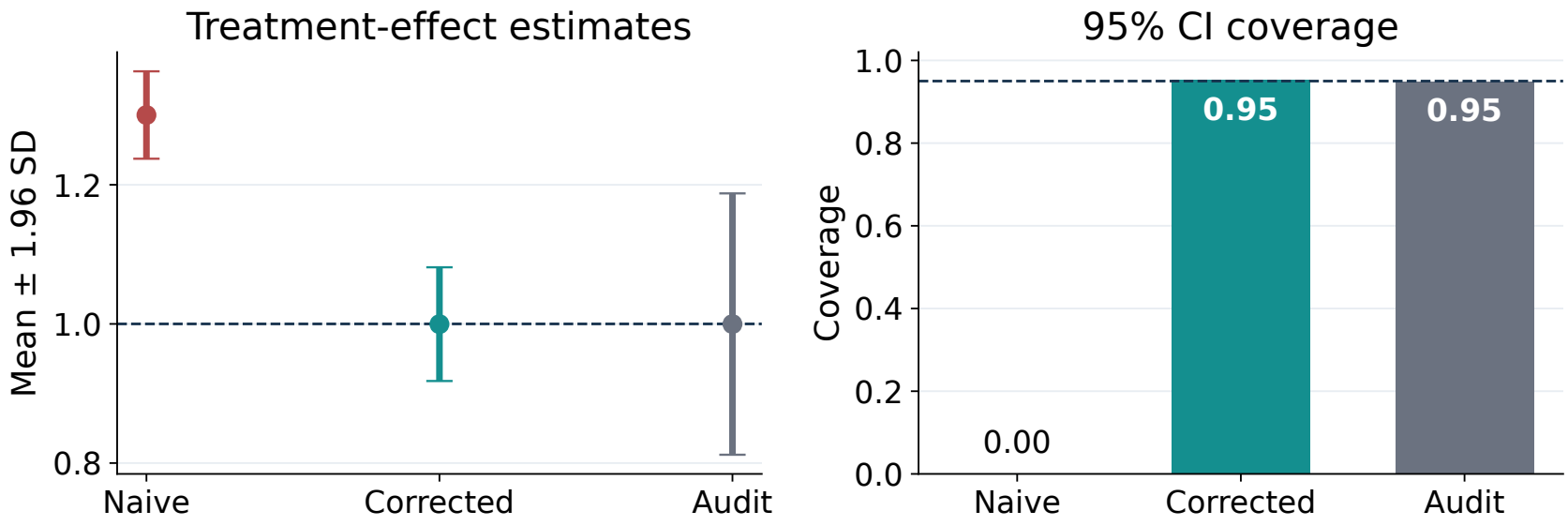
Often, expert labels exist at a granular level, while the variable used in the analysis is aggregated:

$$Y_{it}^{*} \longrightarrow \overline{Y}_{t}^{*} = \frac{1}{N_t} \sum_{i} Y_{it}^{*} \longrightarrow h(\overline{Y}_{t}^{*}).$$

For example, classify individual newspaper articles, aggregate them into an annual EPU index, take logs, then use the index in a regression.

MAR-S propagates the debiasing and uncertainty through these steps.

Back to the simulation



In this linear treatment-contrast example (10% expert audit), Ludwig, DSL, PPI, and MAR-S implement the same AI-plus-expert correction. Their distinctions matter for other estimands, sampling designs, and efficiency.

Some common questions

- If these methods allow the AI/ML model to perform poorly, should I still care about its performance?

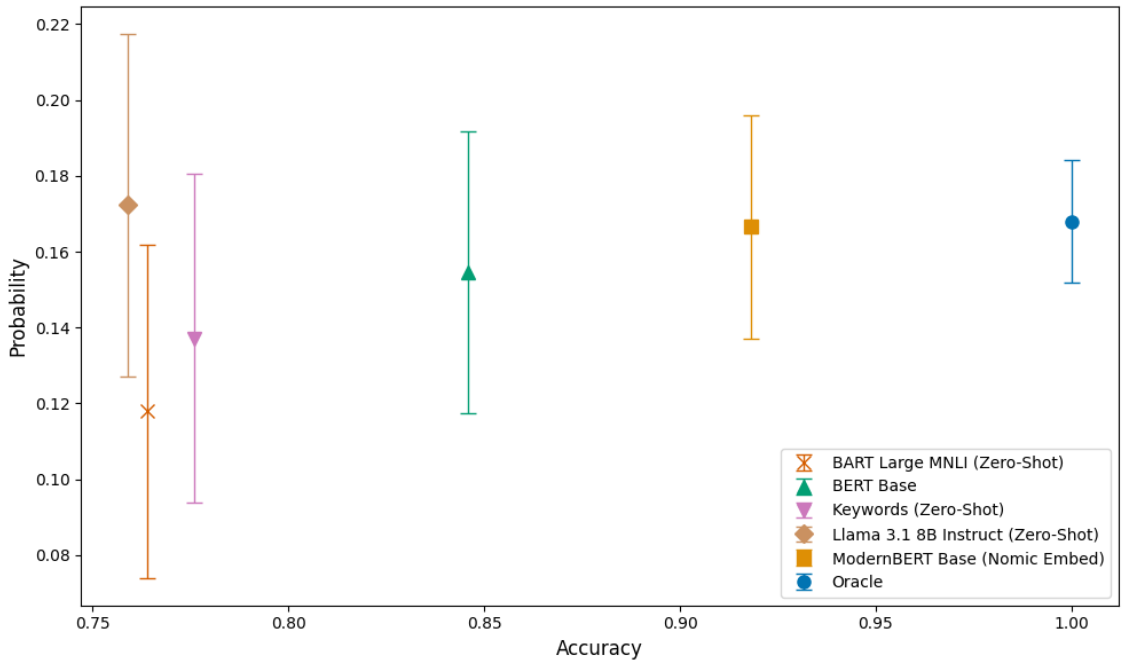
  Yes. Schematically, the more accurate the model, the more precise the estimates.

- If we collect validation data anyway, should we even bother with the possibly mismeasured AI/ML-generated labels on the primary sample?

  Yes. The bias-corrected regression coefficient can be more precisely estimated than the validation-sample-only regression coefficient if the AI/ML-generated labels are sufficiently accurate.

MAR-S with classifiers of varying accuracy (Carlson and Dell, 2026)

Carlson and Dell estimate the share of local U.S. newspaper articles that discuss politics. A random subsample is annotated ($n = 4{,}157$; 20% in total, split evenly between training and debiasing); the remaining annotations are held out, so the corrected estimates can be compared to an oracle using only human labels.



Zero-shot classifiers (keywords, BART MNLI, Llama 3.1 8B) versus encoders fine-tuned on the annotations (BERT, ModernBERT). More accurate classifiers deliver narrower intervals. The oracle observes every true label, including the held-out ones, and is infeasible. Source: Figure 5.

Software

- **PPI:** `ppi_py` (Python), `pip install ppi-python`. Means, quantiles, linear and logistic regression; includes a prediction-powered bootstrap. (`github.com/aangelopoulos/ppi_py`)
- **DSL:** `dsl` (R), from GitHub: `devtools::install_github("naoki-egami/dsl")`. Built for LLM annotations; design-based moments for means, linear, and logistic regression. (`naokiegami.com/dsl`)
- **MAR-S:** `mars` (R), from GitHub: `remotes::install_github("jscarlson/mar-s")`. Means, regression, IV, and causal estimands, plus a generic GMM solver for custom moments.
- Ludwig et al.'s two-regression debiasing needs no package: two OLS fits and a bootstrap.

Outline

Generated regressors arise in two different settings.

We would like to estimate

$$Y_i = \gamma' \theta_i + \alpha' q_i + \varepsilon_i, \qquad \mathbb{E}[\varepsilon_i \mid \theta_i, q_i] = 0,$$

but θ_i is not observed. Two cases:

Supervised learning: θ_i is a well-defined variable that could be observed through human labeling. An AI/ML model predicts $\widehat{\theta}_i = \widehat{g}(U_i)$. For example, whether a job posting offers remote work.

Unsupervised learning: θ_i is genuinely latent and defined through a model for U_i. The algorithm estimates a topic share, latent type, or embedding. For example, CEO work style inferred from a time-use diary.

In both cases, the naive two-step strategy replaces θ_i with $\widehat{\theta}_i$. The correction approach differs in the two settings, but the source of the bias is similar.

Core intuition: the information about θ_i is fixed per document. The measurement error in $\widehat{\theta}_i$ is set by document length, not by sample size.

Collecting more documents:

- shrinks sampling error: averages over documents concentrate;
- does nothing to measurement error: you accumulate more noisy $\widehat{\theta}_i$'s, but not better ones.

So confidence intervals tighten around a biased pseudo-true value.

Battaglia et al.'s theoretical experiment: measurement error improves with n but stays comparable to sampling error, so the estimator is consistent yet conventional inference remains wrongly centered.

The next slides formalize this and propose several debiasing approaches.

Where does generated-regressor bias enter OLS?

Abstracting from covariates and the intercept, if $Y_i = \gamma\theta_i + \varepsilon_i$ but OLS uses $\widehat{\theta}_i$, then

$$\widehat{\gamma} - \gamma = \underbrace{\frac{\sum_i \widehat{\theta}_i \varepsilon_i}{\sum_i \widehat{\theta}_i^2}}_{\text{sampling error}} - \underbrace{\gamma \frac{\sum_i \widehat{\theta}_i (\widehat{\theta}_i - \theta_i)}{\sum_i \widehat{\theta}_i^2}}_{\text{generated-regressor bias}}.$$

Battaglia et al. study the case where the second term's numerator follows:

$$\sqrt{n}\,\mathbb{E}\big[\widehat{\theta}_i(\widehat{\theta}_i - \theta_i)\big] \longrightarrow \kappa\Omega.$$

κ sizes the measurement error relative to sampling error: if $\kappa = 0$, the error moment vanishes faster than $1/\sqrt{n}$ and OLS is fine; if $\kappa > 0$, it is of the same order as sampling noise and the bias survives in large samples. Ω describes how the first-stage measurement error enters the downstream bias.

The general result (Battaglia et al., 2025).

Let $\psi = (\gamma', \alpha')'$, $\xi_i = (\theta_i', q_i')'$, $\widehat{\xi}_i = (\widehat{\theta}_i', q_i')'$, and $Q = \mathbb{E}[\xi_i \xi_i']$. Let Ω describe the shape of the first-stage error and let

$$D = \begin{bmatrix} \Omega & 0 \\ 0 & 0 \end{bmatrix}.$$

Under their regularity and orthogonality conditions,

$$\sqrt{n}(\widehat{\psi} - \psi) \xrightarrow{d} \mathcal{N}\left(-\kappa Q^{-1} D\psi, V\right), \qquad \widehat{V}_{\mathrm{OLS}} \xrightarrow{p} V.$$

The usual heteroskedasticity-robust (sandwich) OLS variance estimator consistently estimates V, so conventional standard errors get the width right.

When $\kappa D\psi \neq 0$, the center is shifted by $-\kappa Q^{-1} D\psi$: the generated regressor distorts the coefficient at the same order as sampling error.

The analytical correction estimates the location shift.

Let

$$\widehat{Q} = \frac{1}{n}\sum_{i=1}^{n} \widehat{\xi}_i \widehat{\xi}_i', \qquad \widehat{D} = \begin{bmatrix} \widehat{\Omega} & 0 \\ 0 & 0 \end{bmatrix}.$$

The estimated asymptotic bias and the additive correction are

$$\boxed{\widehat{b} = -\widehat{\kappa}\widehat{Q}^{-1}\widehat{D}\widehat{\psi}, \qquad \widehat{\psi}^{\mathrm{bc}} = \widehat{\psi} - \frac{\widehat{b}}{\sqrt{n}}.}$$

Because robust OLS standard errors estimate the correct asymptotic variance, the confidence interval is recentered at $\widehat{\psi}^{\mathrm{bc}}$.

Limitation: the correction is first order in the measurement error, so it is valid when measurement error is comparable to sampling error. With large misclassification (short documents, noisy classifiers), it under-corrects; joint estimation remains valid in such settings (see what follows).

What is κ?

Supervised: For the binary-label specialization, $\widehat{\Omega} = 1$ and

$$\widehat{\kappa} = \sqrt{n}\,\widehat{\mathbb{P}}(\widehat{\theta}_i = 1, \theta_i = 0).$$

If this joint false-positive probability is not known externally, it can be estimated using a random audit. Interestingly, only predicted positives in the audit need to be inspected.

Unsupervised: Battaglia et al.'s leading example is a topic model: document i's C_i words are drawn from a mixture over topics with weights θ_i. Estimation noise in $\widehat{\theta}_i$ scales with $1/C_i$ (the information per document), so

$$\widehat{\kappa} = \sqrt{n} \times \Big(\frac{1}{n}\sum_{i=1}^{n} C_i^{-1}\Big),$$

while $\widehat{\Omega}$ is computed from the fitted topic matrix and topic weights. Interestingly, no human labels are required, but the correction is specific to the assumed latent-variable model.

▸ Derivation

Joint estimation is the model-based alternative.

Specify a joint model for outcome, measurement, and latent regressor:

$$\mathcal{L}(\psi, \zeta) = \prod_{i=1}^{n} \int \underbrace{p(Y_i \mid \theta_i, q_i; \psi)}_{\text{outcome model}} \underbrace{p\{g(U_i) \mid \theta_i; \zeta\}}_{\text{measurement model}} \underbrace{p(\theta_i \mid v_i; \zeta)}_{\text{latent distribution}} \, d\theta_i$$

The approach integrates out θ_i and allows Y_i to help infer it. It does not rely on the small-measurement-error approximation. The price: a correctly specified and identified joint model, and more computation.

▸ Why it works

Joint estimation: the supervised case.

$g(U_i) = \widehat{\theta}_i$, and the measurement model describes misclassification. With binary θ_i, the integral becomes

$$\sum_{t \in \{0,1\}} p(Y_i \mid \theta_i{=}t, q_i; \psi)\, p(\widehat{\theta}_i \mid \theta_i{=}t; \zeta)\, p(\theta_i{=}t \mid v_i; \zeta) :$$

a mixture over the two possible true labels. The misclassification model and the prevalence model $p(\theta_i \mid v_i; \zeta)$ supply the weights.

The sum is closed form, so the likelihood can be maximized directly.

Joint estimation: the unsupervised case.

Consider the CEO time-use application. Y_i is log firm sales, $\theta_i \in [0,1]$ is CEO i's latent leadership index, $x_i = (x_{i1}, \ldots, x_{iV})'$ records counts of V activity types across $C_i = \sum_j x_{ij}$ surveyed 15-minute slots, and β_0, β_1 are the "management" (e.g., visit production sites) and "leadership" (e.g., interactions with senior executives) activity distributions. v_i predicts CEO behavior and q_i contains controls in the sales regression.

The joint model is

$$\begin{aligned}
\text{logit}(\theta_i) \mid v_i &\sim \mathcal{N}\left(\varphi' v_i, \sigma_\theta^2\right), && \text{(latent behavior)} \\
x_i \mid \theta_i, C_i &\sim \text{Multinomial}(C_i, (1-\theta_i)\beta_0 + \theta_i \beta_1), && \text{(time-use diary)} \\
Y_i \mid \theta_i, q_i &\sim \mathcal{N}\left(\gamma\theta_i + \alpha' q_i, \sigma^2\right). && \text{(sales regression)}
\end{aligned}$$

No human labels are required. Identification comes from the assumed latent-variable model. The integral over θ_i has no closed form, so Battaglia et al. use Hamiltonian Monte Carlo in NumPyro.

▸ Likelihood

Battaglia et al.'s generated-labels simulation

$Y_i = 10 + \theta_i + (0.3 + 0.2\theta_i)\varepsilon_i$, $\theta_i \sim \text{Bernoulli}(0.025)$, misclassification scaled so that $\kappa \in \{0.5, 1, 2\}$; the corrections use an audit of $m = 1{,}000$ labels.

Table 4: Simulation Results: Generated Labels, $p = 0.025$

	Bias			RMdSE			Coverage		
κ	0.5	1	2	0.5	1	2	0.5	1	2
				$n = 8000$					
2-Step	-0.228	-0.459	-0.916	0.228	0.459	0.916	0.001	0.000	0.000
BCA-1	-0.055	-0.214	-0.843	0.076	0.214	0.843	0.899	0.589	0.000
BCA-2	-0.039	-0.203	-0.841	0.073	0.204	0.841	0.929	0.643	0.000
BCM-1	-0.003	-0.006	-0.715	0.091	0.170	0.806	0.887	0.758	0.265
BCM-2	0.023	0.030	-0.729	0.090	0.177	0.827	0.904	0.773	0.257
Joint	0.000	0.002	-0.008	0.045	0.056	0.111	0.935	0.930	0.817

BCA/BCM: additive/multiplicative corrections; -1 uses the audit's empirical false-positive rate, -2 a Bayes estimate of it ("FPR" is the joint probability $\Pr(\widehat{\theta}_i = 1, \theta_i = 0)$, not the conditional rate). Two-step coverage is zero; the corrections work for moderate κ and improve with n (panels for $n = 16{,}000$ and 32,000 in the paper); joint estimation performs best and attains nominal coverage as n grows. Source: Table 4, $n = 8{,}000$ panel.

Battaglia et al.'s generated-indices simulation

$\theta_i \sim U[0,1]$, $N_i \sim \text{Binomial}(C_i, 0.1 + 0.8\theta_i)$, $Y_i = -0.05 + 0.11\theta_i + 0.1\varepsilon_i$; document lengths C_i scaled so that $\kappa \in \{4.57, 2.28, 1.14\}$.

Table 7: Simulation Results: AI/ML-Generated Indices

	Bias			RMdSE			Coverage		
κ	4.57	2.28	1.14	4.57	2.28	1.14	4.57	2.28	1.14
				$n = 200$					
2-Step	-0.446	-0.299	-0.180	0.049	0.033	0.022	0.309	0.677	0.839
BCA	-0.148	-0.019	0.018	0.026	0.021	0.019	0.716	0.813	0.880
BCM	0.240	0.183	0.081	0.040	0.029	0.021	0.495	0.678	0.844
Joint	-0.003	0.007	0.004	0.024	0.020	0.018	0.945	0.948	0.938
2-Step-Share	-0.433	-0.218	-0.037	0.048	0.025	0.018	0.378	0.824	0.931

Bias is median *relative* bias: at $\kappa = 4.57$ the two-step estimate misses γ by nearly half. 2-Step-Share regresses on the raw share N_i / C_i. The corrections remove most bias; only joint estimation restores nominal coverage (larger n in the paper). Source: Table 7, $n = 200$ panel.

Software: the generated variable as a regressor

- **Analytical bias corrections** (Battaglia et al.): `ValidMLInference` (Python, on PyPI) and `MLBC` (R, on CRAN).
- **Joint estimation:** `NumPyro` (Python), Hamiltonian Monte Carlo for the joint measurement-plus-outcome likelihood.

Outline

Recap: the generated variable as an outcome

- Bias arises when measurement errors correlate with the regressors:

$$\tau_{\text{naive}} = \tau + \mathbb{E}[e_i \mid D_i = 1] - \mathbb{E}[e_i \mid D_i = 0].$$

- High predictive accuracy of the AI/ML model does not guarantee valid inference (but it helps with CIs).

- The fix: collect ground-truth labels for a random subsample, then impute and correct, at the level of the coefficient (Ludwig et al.), the outcome (DSL), the estimating score (PPI), or the influence function (MAR-S).

- Valid for *any* predictor; accuracy buys precision; the labeling sampling design is potentially yours to choose when you collect the annotations.

Recap: the generated variable as a regressor

- When measurement error is large relative to sampling error, confidence intervals have the right width but the wrong center.
- The correction requires a measurement model: an audited false-positive rate for generated labels or a "structural" latent-variable model.
- Two routes: subtract the estimated asymptotic bias (analytical correction), or integrate out the latent regressor by estimating the measurement and outcome models jointly. The latter performs best.
- (You can also still use MAR-S.)

Additional interesting references

- Modarressi et al. (2025): LLM-learned outcomes in randomized experiments, combining sample splitting with human validation.
- Rambachan et al. (2024): treatment effects when the outcome is measured only by remote sensing (satellite imagery, phone activity).
- Christensen et al. (2026): bootstrap inference when AI/ML-generated labels enter as regressors.
- Christensen and Compiani (2026): demand counterfactuals when product attributes are extracted from unstructured data with error.
- Imai and Nakamura (2026): the text itself as the *treatment*, using generative AI to construct counterfactual texts.

References I

Angelopoulos, A. N., Bates, S., Fannjiang, C., Jordan, M. I., and Zrnic, T. (2023). Prediction-powered inference. *Science*, 382(6671):669–674.

Ash, E. and Hansen, S. (2023). Text algorithms in economics. *Annual Review of Economics*, 15:659–688.

Battaglia, L., Christensen, T., Hansen, S., and Sacher, S. (2025). Inference for regression with variables generated by ai or machine learning. arXiv:2402.15585, version 5, April 2025.

Carlson, J. and Dell, M. (2026). A unifying framework for robust and efficient inference with unstructured data. arXiv:2505.00282, version 3, February 2026.

Christensen, T. and Compiani, G. (2026). From unstructured data to demand counterfactuals: Theory and practice. arXiv:2601.05374.

Christensen, T., Goncalves, S., and Perron, B. (2026). Bootstrapping with ai/ml-generated labels. arXiv:2604.23770.

Dell, M. (2025). Deep learning for economists. *Journal of Economic Literature*, 63(1):5–58.

Egami, N., Hinck, M., Stewart, B. M., and Wei, H. (2023). Using imperfect surrogates for downstream inference: Design-based supervised learning for social science applications of large language models. In *Advances in Neural Information Processing Systems*, volume 36.

Gentzkow, M., Kelly, B., and Taddy, M. (2019). Text as data. *Journal of Economic Literature*, 57(3):535–574.

Imai, K. and Nakamura, K. (2026). Causal inference with generative artificial intelligence: Application to texts as treatments. *Journal of the American Statistical Association*.

References II

Ludwig, J., Mullainathan, S., and Rambachan, A. (2026). Large language models: An applied econometric framework. *Annual Review of Economics*, 18:283–316.

Modarressi, I., Spiess, J., and Venugopal, A. (2025). Causal inference on outcomes learned from text. arXiv:2503.00725.

Rambachan, A., Singh, R., and Viviano, D. (2024). Program evaluation with remotely sensed outcomes. arXiv:2411.10959.

Backup slides

Why does it "work"?

Under the sampling design, $Y_i^* \perp R_i \mid V_i$, so

$$\begin{aligned}\mathbb{E}[\widetilde{Y}_i \mid V_i] &= \widehat{m}(V_i) + \frac{\mathbb{E}[R_i \mid V_i]}{\pi(V_i)} \{\mathbb{E}[Y_i^* \mid V_i] - \widehat{m}(V_i)\} \\ &= \widehat{m}(V_i) + \mathbb{E}[Y_i^* \mid V_i] - \widehat{m}(V_i) \\ &= \mathbb{E}[Y_i^* \mid V_i].\end{aligned}$$

$\widehat{m}$ cancels, this is true for any $\widehat{m}$, so it may be arbitrarily biased and the estimator remains valid!

◂ Back

Concretely, how does DSL construct $m(V_i)$?

In the running example, V_i could contain the AI label, the document text, document length, treatment status, and unit characteristics.

DSL uses cross-fitting:

1. Split the documents into K folds.
2. For fold k, use the expert-labeled documents outside that fold to fit

$$\widehat{m}^{(-k)}(V_i) \approx \mathbb{E}[Y_i^* \mid V_i].$$

3. For each document i in fold k, construct

$$\widetilde{Y}_i = \widehat{m}^{(-k)}(V_i) + \frac{R_i}{\pi(V_i)}\{Y_i^* - \widehat{m}^{(-k)}(V_i)\}.$$

4. Regress $\widetilde{Y}_i$ on $W_i = (1, D_i)'$ using all documents.

Why out-of-fold? The previous slide treated $\widehat{m}$ as fixed. If document i's own label helped fit $\widehat{m}$, overfitting could bias the correction; out-of-fold prediction rules this out.

Back

PPI for common estimands (Angelopoulos et al., 2023)

Estimand	Prediction-based gradient $\hat{g}_\theta^f$	Rectifier $\hat{\mathbf{\Delta}}_\theta$	Procedure
Mean	$\theta - \frac{1}{N}\sum_{i=1}^{N} f(\widetilde{X}_i)$	$\frac{1}{n}\sum_{i=1}^{n}(f(X_i) - Y_i)$	Alg. 1
Median	$\frac{1}{2N}\sum_{i=1}^{N} \text{sign}(\theta - f(\widetilde{X}_i))$	$\frac{1}{n}\sum_{i=1}^{n}\left(\mathbb{1}\left\{f(X_i) \le \theta\right\} - \mathbb{1}\left\{Y_i \le \theta\right\}\right)$	Alg. 2
q-quantile	$-q + \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left\{f(\widetilde{X}_i) \le \theta\right\}$	$\frac{1}{n}\sum_{i=1}^{n}\left(\mathbb{1}\left\{f(X_i) \le \theta\right\} - \mathbb{1}\left\{Y_i \le \theta\right\}\right)$	Alg. 2
Logistic regression	$\frac{1}{N}\sum_{i=1}^{N} \widetilde{X}_i^T\left(\frac{1}{1+e^{-\hat{\theta}^T\widetilde{X}_i}} - f(\widetilde{X}_i)\right)$	$\frac{1}{n}\sum_{i=1}^{n} X_i(f(X_i) - Y_i)$	Alg. 3
Linear regression	$\theta - \widetilde{X}^T f(\widetilde{X})$	$X^+(f(X) - Y)$	Alg. 4
Convex minimizer	$\frac{1}{N}\sum_{i=1}^{N} \nabla\ell_\theta(\widetilde{X}_i, f(\widetilde{X}_i))$	$\frac{1}{n}\sum_{i=1}^{n}(\nabla\ell_\theta(X_i, f(X_i)) - \nabla\ell_\theta(X_i, Y_i))$	Alg. 5

Table 1: Prediction-powered inference for common statistical problems.

Their notation: $f(X_i)$ is the AI label (our $\widehat{Y}_i$), Y_i the ground-truth label (our Y_i^*), and $\widetilde{X}_i$ runs over the unlabeled sample. Up to a sign convention, the "prediction-based gradient" is the score evaluated at the AI labels (our imputation term) and the rectifier is our correction.

◂ Back

What is the imputation term $\mathbb{E}[\phi_i^{\text{full}}(\theta) \mid V_i]$?

It replaces every unobserved quantity in ϕ_i^{full} by its best prediction given the observables.

For the mean, $\theta^* = \mathbb{E}[Y_i^*]$:

$$\phi_i^{\text{full}}(\theta) = Y_i^* - \theta \quad \Longrightarrow \quad \mathbb{E}[\phi_i^{\text{full}}(\theta) \mid V_i] = \mathbb{E}[Y_i^* \mid V_i] - \theta.$$

Estimating $\mathbb{E}[Y_i^* \mid V_i]$ with $\widehat{m}(V_i)$, the observed-data influence function is the centered pseudo-outcome:

$$\widehat{\phi}_i^{\text{MAR-S}}(\theta) = \widetilde{Y}_i - \theta.$$

Back

Why do false positives appear in the binary formula?

The leading OLS error term is

$$\widehat{\theta}_i(\widehat{\theta}_i - \theta_i).$$

For a hard binary label, its value in each cell of the confusion matrix is

θ_i	$\widehat{\theta}_i$	$\widehat{\theta}_i(\widehat{\theta}_i - \theta_i)$
1	1 (true positive)	0
0	1 (false positive)	1
1	0 (false negative)	0
0	0 (true negative)	0

False negatives vanish here because OLS multiplies the error by the *observed* regressor: $\widehat{\theta}_i = 0$. Randomized hard labels leave the same leading false-positive moment.

◂ Back

The CEO time-use likelihood

The observed-data likelihood integrates out θ_i:

$$\mathcal{L}(\gamma, \alpha, \sigma, \beta, \zeta) \propto \prod_{i=1}^{n} \int_0^1 \frac{1}{\sigma}\, \phi\!\left(\frac{Y_i - \gamma\theta - \alpha' q_i}{\sigma}\right) \prod_{j=1}^{V} \lambda_j(\theta)^{x_{ij}}\, p(\theta \mid v_i; \zeta)\, d\theta,$$

with $\lambda_j(\theta) = (1-\theta)\beta_{0j} + \theta\beta_{1j}$ the probability that a slot shows activity type j under leadership weight θ, and a logistic-normal $p(\theta \mid v_i; \zeta)$ whose mean shifts with log employment and an MBA indicator.

Back

Why does joint estimation work for a regressor?

The two-step regression replaces all uncertainty about θ_i by a point:

$$p(\theta_i \mid U_i) \quad \longrightarrow \quad \widehat{\theta}_i.$$

This creates errors-in-variables bias. The joint likelihood instead averages the outcome model over every plausible value:

$$p(Y_i, U_i \mid q_i, v_i) = \int p(Y_i \mid \theta_i, q_i; \psi) p(U_i \mid \theta_i; \zeta) p(\theta_i \mid v_i; \zeta)\, d\theta_i.$$

For a remote-work label, a posting can contribute fractionally to the remote and non-remote outcome distributions according to its misclassification probabilities. For the CEO example, a short diary produces a diffuse range of leadership shares, while a long diary produces a concentrated range. The regression is estimated over those ranges rather than on noisy point scores. This removes plug-in measurement-error bias when the measurement, latent, and outcome models are correctly specified and jointly identified. Integration is the correction; HMC is only a way to compute it.

Back