Swiss Federal Institute of Technology Zurich

Seminar for Statistics

Department of Mathematics

Master Thesis Summer 2025

Roger Li

Statistical Inference Using Prediction Powered Inference and Predict-Then-Debias

Submission Date: 08 September 2025

Advisor: Dr. Markus Kalisch

Abstract

Prediction-powered inference (PPI) combines noisy proxy data with limited high-quality labels to deliver valid statistical inference. This thesis reviews PPI, its efficient extension PPI++, and the Predict-Then-Debias (PTD) estimator, which broadens applicability to settings where both covariates and outcomes may be measured with error.

Through simulation studies in linear and spatial settings, both PPI++ and PTD improved confidence interval width relative to gold-standard-only inference while preserving empirical coverage close to theoretical guarantees. PPI++ was most effective when error was confined to the outcome, producing consistently narrower intervals, whereas PTD provided robustness when covariates were also corrupted.

Applications reinforced these insights: in a wine quality dataset, PPI++ detected additional significant predictors without false positives, and in a forest cover dataset, PTD produced estimates closer to the true parameters with tighter intervals.

Overall, PPI++ is recommended when covariates are measured accurately and only the response is noisy, while PTD is better suited to more general error structures. These results underscore the promise of prediction-powered inference as a practical tool for research scenarios where abundant but imperfect data coexist with scarce gold-standard observations.

Contents

Chapter 1

Introduction

1.1 Introduction

In the present day, collecting large amounts of data is relatively easy, but ensuring the accuracy of this data is much harder. For purely predictive tasks, imperfect accuracy may not be a major issue, since many machine learning models are robust to noisy inputs. However, when the goal is statistical inference, accuracy matters greatly. Confidence intervals and hypothesis tests are only valid when their assumptions hold, and noisy or biased data can quickly break those guarantees.

This tension is especially visible in geo-spatial applications. Satellite imagery provides abundant data, but such measurements are often indirect and can suffer from measurement error or systematic bias. By contrast, manually collected field-survey data is highly accurate, but expensive and time-consuming to gather (Bright et al., 2020). Similar issues occur outside of remote sensing: in machine learning applications with scarce labeled data, in clinical studies with limited access to expert-verified outcomes, in epidemiology with hard-to-measure exposures, and in survey analysis where self-reported answers may be unreliable. In all of these cases, we often have a large supply of imperfect proxy data, and a small amount of accurate but costly ground-truth data.

Recently, new methods have been developed to address this problem. Prediction-powered inference (PPI) (Angelopoulos et al., 2023) and its extension PPI++ (Angelopoulos et al., 2024) allow researchers to use large amounts of proxy data for the outcome variable Y, combined with a smaller set of gold-standard labels. The Predict-Then-Debias (PTD) estimator (Kluger et al., 2025) generalizes this idea further, allowing both covariates X and outcomes Y to be measured with error. Whereas PPI and PPI++ correct bias by modifying the loss function before estimation (loss-debiasing), PTD first estimates the parameter of interest and then applies a bias correction step (prediction-debiasing).

In this thesis, I will use the terms *map-product* and *remotely-sensed* data interchangeably, to mean imperfect data that serves as a useful but biased proxy for a true variable. Similarly, the terms *gold-standard* data and *ground-truth* data both refer to accurate observations, usually collected manually by experts, and are therefore costly to obtain. Additionally, all code used to generate results in this paper is accessible via a GitHub repository

1.2 Related Work

The problem of combining noisy proxy data with limited high-quality ground truth has been studied in several domains. In statistics, methods for measurement error correction and debiasing have a long history, though many approaches are tailored to specific models or sampling designs. For example, Chen and Chen (2000) propose a prediction-debiasing method for consistent inference under generalized linear models, but their approach requires double-sampling and is not easily generalized beyond GLMs. Various other prediction-debiasing approaches are limited in a similar fashion, such as Zhang et al. (2019) which focuses solely on mean estimation for semi-supervised data.

Prediction-powered inference methods stand out because of their simplicity and flexibility. PPI++ and PTD can be applied across a range of models and parameters without heavy tailoring, making them broadly useful. Applications are beginning to appear in several areas: Lu et al. (2025) demonstrate the use of satellite-derived data for regression coefficient estimation, Poulet et al. (2025) apply PPI++ to estimate average treatment effects in clinical studies, and Einbinder et al. (2025) extend PPI to construct risk-controlling prediction sets that help calibrate machine learning models.

Chapter 2

Prediction-Powered Inference

2.1 Problem Setting

Suppose we have access to a small, high-quality (gold-standard) dataset

$$(X, Y) = \{(X_1, Y_1), \ldots, (X_n, Y_n)\} \overset{\text{i.i.d.}}{\sim} \mathbb{P},$$

which is accurate but expensive to collect. In addition, we possess a much larger dataset, but potentially biased or noisy in Y,

$$(\tilde{X}, \tilde{Y}) = \{(\tilde{X}_1, \tilde{Y}_1), \ldots, (\tilde{X}_N, \tilde{Y}_N)\} \overset{\text{i.i.d.}}{\sim} \tilde{\mathbb{P}},$$

with $N \gg n$, and $\mathbb{P}_X = \tilde{\mathbb{P}}_{\tilde{X}}$. A typical example would involve (X, Y) obtained from survey data, while $(\tilde{X}, \tilde{Y})$ comes from a map product or sensor.

Our goal is to estimate a quantity of interest, $\theta \in \mathbb{R}^d$, such as a population mean or an OLS coefficient, and construct a valid confidence interval. Traditionally, there are two approaches for inference in this setting:

i.) **Naive Estimation:** Use the noisy dataset as if it were ground truth. This can lead to biased or invalid inference.

ii.) **Gold-Standard Only:** Discard the noisy data and rely solely on the gold-standard dataset, yielding valid but inference limited by sample size.

Prediction-Powered Inference (PPI), introduced by Angelopoulos et al. (2023), offers a third approach. It uses both gold-standard and noisy data to perform statistically valid inference without wasting resources. Specifically, PPI uses predictions from a black-box model and corrects the bias using the gold-standard data, resulting in tighter, statistically sound confidence intervals.

First, we define the *rectifier*, Δ, a function that adjusts for the bias in the black-box predictions, $f(\tilde{X})$,using the gold-standard estimate. Using the rectifier and the gold-standard dataset (X, Y), we construct a *Rectifier Confidence Set*, denoted by $\mathcal{R}$. Finally, the Prediction-Powered Confidence Set, $\mathcal{C}^{PP}$, is formed by applying the rectifier to the black-box estimate $\hat{\theta}_f$. Both PPI and its enhanced version, PPI++, follow this framework, with PPI++ introducing modifications to the confidence interval procedure.

Key definitions:

- **Parameter of Interest:** θ
- **Gold-Standard Estimate:** $\hat{\theta} = \hat{\theta}(X, Y)$
- **Black-Box Estimate:** $\hat{\theta}_f = f(\tilde{X})$
- **Rectifier:** $\Delta = \Delta(\theta, \theta_f)$
- **Empirical Rectifier:** $\hat{\Delta} = \hat{\Delta}(\hat{\theta}, \hat{\theta}_f)$
- **Prediction-Powered Estimate:** $\hat{\theta}^{PP} = \hat{\theta}^{PP}(\hat{\theta}_f, \hat{\Delta})$

2.2 Prediction-Powered Inference (PPI)

PPI enables black-box predictions to supplement ground-truth data for valid statistical analysis. When the black-box model is reasonably accurate, the method should yield confidence intervals that are both valid and more efficient.

The parameter of interest is defined as:

$$\theta = \arg\min_{\theta} \mathbb{E}[\ell_\theta(X, Y)],$$

where ℓ_θ is a loss function associated with θ.

2.2.1 Convex Estimation

Assuming convexity in ℓ_θ allows us to reformulate θ^* as the solution of:

$$\mathbb{E}[g_\theta(X, Y)] = 0,$$

where $g_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^p$ is the subgradient of ℓ_θ with respect to θ.

The rectifier is defined as:

$$\Delta_\theta = \mathbb{E}[g_\theta(X, Y) - g_\theta^f(X, f(X))]$$

where $f(X)$ denotes the black-box prediction.

We then construct a confidence set $\mathcal{R}_\delta(\theta)$ such that:

$$P\left(\Delta_\theta \in \mathcal{R}_\delta(\theta)\right) \geq 1 - \delta$$

and form a confidence set for θ^* by going through all values of θ, and finding a confidence set $\mathcal{T}_{\alpha-\delta}(\theta)$ that satisfies

$$P(g_\theta^f \in \mathcal{T}_{\alpha-\delta}(\theta)) \geq 1 - (\alpha - \delta)$$

Finally, using everything from before we can construct the prediction-powered confidence interval [1]

$$\mathcal{C}^{PP} = \{\theta : \mathcal{R}_\delta(\theta) + \mathcal{T}_{\alpha-\delta}(\theta)\}$$

with $P(\theta^* \in \mathcal{C}_\alpha^{PP}) \geq 1 - \alpha$.

[1] *Note + in the equation for C^{PP} is the Minkowski Sum*

2.2.2 Non-Convex Estimation

The PPI framework works without the assumption of convex loss, but there is a loss in computational efficiency and power. You can see the derivation and form of this estimator in Angelopoulos et al. (2023), however PPI++ renders much of PPI obsolete in use.

2.3 Efficient Prediction-Powered Inference (PPI++)

Efficient Prediction-Powered Inference (PPI++), introduced by Angelopoulos et al. (2024), enhances the original PPI framework by improving computational scalability and extending its applicability to areas such as generalized linear models (GLMs). Notably, PPI++ no longer requires the black-box model to be accurate in order to be at least as good as the naive estimator.

PPI++ is derived similarly to PPI, but with one key conceptual change. PPI aims to estimate a confidence interval for each possible θ, which is computationally infeasible, thus requiring special simplifications to be applied effectively. PPI++ instead derives an asymptotically normal estimate of θ via CLT, which is much simpler.

Specifically, the resulting formulation of $\mathcal{C}^{PP}$ changes as follows:

$$\textbf{PPI:} \quad \mathcal{C}^{PP} = \left\{\theta : 0 \in \mathcal{C}_\alpha^\Delta\right\}$$

$$\textbf{PPI++:} \quad \mathcal{C}^{PP} = \left\{\hat{\theta}^{PP} \pm z_{1-\frac{\alpha}{2}} \frac{\hat{\sigma}}{\sqrt{n}}\right\}$$

2.3.1 PPI++ Theory

To begin deriving the general form of a PPI++ confidence interval, define the standard loss using gold-standard data $L(\theta)$, black-box model loss using gold-standard data $L^f(\theta)$, and the black-box model loss using map product data $\tilde{L}^f(\theta)$. Like before, these aim to quantify the differences between our black-box model and standard model by comparing the results when using gold-standard and map product data.

$$\underbrace{L(\theta) := \mathbb{E}[\ell_\theta(X, Y)],}_{\text{ground-truth loss}} \quad \underbrace{L^f(\theta) := \mathbb{E}[\ell_\theta(X, f(X)],}_{\text{loss of black-box model using ground truth data}} \quad , \quad \underbrace{\tilde{L}^f(\theta) := \mathbb{E}[\ell_\theta(\tilde{X}, f(\tilde{X})],}_{\text{naive loss}}$$

With the goal of quantifying the accuracy of the PPI estimate, we define the *rectified loss*:

$$L^{PP}(\theta) := \underbrace{\tilde{L}_N^f(\theta)}_{\text{biased naive loss}} - \underbrace{(L_n^f(\theta) - L_n(\theta))}_{\text{bias in loss from } f}$$

which is the ground-truth loss, $L(\theta)$, subtracted by the bias of the black-box prediction loss $(\tilde{L}^f(\theta) - L^f(\theta))$. Finally we can define the prediction-powered point-estimate, $\hat{\theta}^{PP}$, and corresponding confidence interval, $\mathcal{C}^{PP}$,

$$\hat{\theta}^{PP} = \arg\min_\theta L^{PP}(\theta), \quad \mathcal{C}^{PP} = \left\{\hat{\theta}^{PP} \pm z_{1-\frac{\alpha}{2}} \frac{\hat{\sigma}}{\sqrt{n}}\right\}$$

where σ^2 is a consistent estimator of the variance of $\hat{\theta}^{PP}$

Significant simplifications for computation are admitted on a model-to-model basis. Below, I only cover PPI++ for GLMs, however the original paper covers algorithms for exponential-family models and for M-estimators with [2]*smooth enough losses*.

2.3.2 PPI++ Power Tuning

An important addition from PPI to PPI++ is included power tuning as a plug-in estimator (*covered in Angelopoulos et al. (2024)*). Recall for PPI to be effective, there is a stipulation that the black-box model f needs to be *accurate enough* to guarantee results are not worse than the classical estimate. PPI+ introduces the tuning parameter $\lambda \in \mathbb{R}$, which scales the bias of the black-box prediction loss in L^{PP}, resulting in

$$\hat{\theta}_{\lambda}^{PP} = \arg\min_{\theta} L_{\lambda}^{PP}, \quad L_{\lambda}^{PP} := L_n(\theta) - \lambda(\tilde{L}_N^f(\theta) + L_n^f(\theta))$$

where $\lambda = 1$ results in the untuned prediction, while $\lambda = 0$ is equivalent to classical inference. In practice, this means that if our black-box prediction is poor, we rely less on it.

2.3.3 Recap of Generalized Linear Models

To exemplify the use cases of PPI++, we first review generalized linear models (GLMs). Recall that a probability distribution is said to belong to an *exponential family* if it can be written in the form

$$p(x|\theta) = h(x)\exp(w(\theta)^{\intercal}t(x) - \psi(\theta)), \quad h(x), c(\theta) \geq 0$$

- $h(x)$ is the *base-measure*
- $w(\theta)$ is the *natural parameter*
- $t(x)$ is a *sufficient statistic* for θ
- $\psi(\theta)$ is the *log-partition function*

Common distributions in the exponential family are as Gaussian, binomial, and exponential. If we're interested in modeling a response variable Y from a distribution in an exponential family, then we can do so using a *generalized linear model* if we have the following:

i.) Y follows a distribution in an exponential family

ii.) $\eta = X\beta$: X is linear in the parameters

iii.) $g(\mathbb{E}(Y|X)) = \eta = \mu$: There is a link function g that explains how the expected value of Y relates to the linear combination of predictors $X\beta$

GLMs as a class of models includes linear regression, logistic regression, and Poisson regression among others.

[2] *smooth enough losses here means locally Lipschitz and differentiable at θ with probability 1*

2.3.4 PPI++ for GLMs

For basic GLMs, we can express the loss as

$$\ell_\theta(x, y) = -\log(p_\theta(y|x)) = -y\theta^\intercal x + \psi(x^\intercal\theta))$$

where ψ is the log-partition function (e.g., $\psi(s) = \frac{1}{2}s^2$ for logistic regression)

Under suitable assumptions, the following algorithm can be used:

Algorithm: PPI++ Confidence Intervals for GLMs

Input: labeled data (X, Y), biased data $(\tilde{X}, f(\tilde{X}))$, predictive model f, error level $\alpha \in (0, 1)$, coefficient index $j \in 1, \dots, p$ of interest

i.) Select tuning parameter $\hat{\lambda}$

ii.) $\hat{\theta}^{\mathrm{PP}}_{\hat{\lambda}} = \arg\min_\theta L^{\mathrm{PP}}_{\hat{\lambda}}(\theta)$

iii.) $\hat{H} = \frac{1}{N+n}\left(\sum_{i=1}^{n} \psi''(X_i^\intercal \hat{\theta}^{\mathrm{PP}}_{\hat{\lambda}}) X_i X_i^\intercal + \sum_{i=1}^{N} \psi''(\tilde{X}_i^\intercal \hat{\theta}^{\mathrm{PP}}_{\hat{\lambda}}) \tilde{X}_i \tilde{X}_i^\intercal\right)$

iv.) $\hat{V}_f = \hat{\lambda}^2 \mathrm{Cov}_{N+n}\left(\left[\psi'(X_i^\intercal \hat{\theta}^{\mathrm{PP}}_{\hat{\lambda}}) - f(X_i)\right] X_i\right)$

v.) $\hat{V}_\Delta = \mathrm{Cov}_n\left(\left[(1-\hat{\lambda})\psi'(X_i^\intercal \hat{\theta}^{\mathrm{PP}}_{\hat{\lambda}}) + (\hat{\lambda} f(X_i) - Y_i)\right] X_i\right)$

vi.) $\hat{\Sigma} = \hat{H}^{-1}\left(\frac{1}{N}\hat{V}_f + \hat{V}_\Delta\right)\hat{H}^{-1}$

Output: prediction-powered confidence interval

$$C^{\mathrm{PP}}_\alpha = \left(\hat{\theta}^{\mathrm{PP}}_{\hat{\lambda},j} \pm z_{1-\alpha/2}\sqrt{\frac{\hat{\Sigma}_{jj}}{n}}\right) \quad \text{for each } j$$

The resulting estimator converges in distribution for large enough n:

$$\sqrt{n}(\hat{\theta}^{PP}_{\hat{\lambda}} - \theta^*) \xrightarrow{d} \mathcal{N}(0, \Sigma^\lambda)$$

This result extends to M-estimators and exponential-family models, as shown in the original paper.

2.4 Predict-Then-Debias (PTD) Estimator

PPI++ estimation is designed for situations where the response variable Y is remotely sensed or a map-product. However, its guarantees do not extend to settings where the covariates X contain measurement error. In such cases, another variant of prediction-powered inference becomes relevant: the Predict-Then-Debias (PTD) estimator (Kluger et al., 2025). PTD is an alternative to PPI++ that allows for error in X, in Y, or in both simultaneously.

As the name suggests, Predict-Then-Debias differs from PPI++ in that it first estimates the parameter of interest and then debiases this estimate afterwards. In contrast, PPI and PPI++ debias via the rectifier before estimating the parameter of interest.

2.4.1 PTD Notation

Since PPI only allows for errors in Y, it handles gold-standard data (X, Y) and noisy data $(\tilde{X}, \tilde{Y})$ separately, clearly differentiating between X and Y throughout the entire process. PTD, by contrast, accommodates error in both X and Y, making its notation inherently more complex. Specifically, PTD focuses on separating variables that are always observed accurately (ground-truth variables) from those that are noisy and have only a few accurate observations (map-product variables), rather than strictly separating X and Y.

For this section, we consider the entire dataset $\mathcal{D} \in \mathbb{R}^{(N+n)\times p}$, where each $D_i = (X_i, Y_i)$ corresponds to the i-th observation.

Let $D := (D^{\text{easy}}, D^{\text{hard}}, \tilde{D}^{\text{hard}}),\ D \in \mathbb{R}^{n\times p}$ denote the gold-standard data, where D^{easy} and D^{hard} are gold-standard samples for easy-to-measure and hard-to-measure variables respectively, and $\tilde{D}^{\text{hard}}$ represents map-product observations for hard-to-measure variables.

Similarly, let $\tilde{D} := (D^{\text{easy}}, \tilde{D}^{\text{hard}}),\ \tilde{D} \in \mathbb{R}^{N\times p}$ denote the map-product data, where D^{easy} is fully observed as before, and $\tilde{D}^{\text{hard}}$ contains only map-product measurements, without the corresponding gold-standard samples for the hard-to-measure variables.

We are interested in estimating the parameter $\theta \in \mathbb{R}^p$ via $\hat{\theta} = \hat{\theta}(D)$. We also define $\hat{\gamma} = \hat{\gamma}(\tilde{D}) = \hat{\theta}(\tilde{D})$ as the estimate obtained using the map-product data, highlighting that it does not rely entirely on gold-standard measurements.

2.4.2 PTD Theory

The Predict-Then-Debias estimator is defined as

$$\hat{\theta}^{PTD} := \underbrace{\hat{\gamma}_N(D_N^{\text{easy}}, \tilde{D}_N^{\text{hard}})}_{\text{biased naive estimate}} \quad - \quad \underbrace{(\hat{\theta}_n(D_n^{\text{easy}}, \tilde{D}_n^{\text{hard}}) - \hat{\gamma}_n(D_n^{\text{easy}}, D_n^{\text{hard}}))}_{\text{bias correction using gold-standard data}}.$$

Under the assumption that the gold-standard data is collected randomly, the variance of the PTD estimator satisfies

$$\text{Var}(\hat{\theta}^{PTD}) = \text{Var}(\hat{\gamma}_N) + \text{Var}(\hat{\theta}_n - \hat{\gamma}_n).$$

Similar to PPI++, PTD incorporates a tuning matrix $\hat{\Omega} \in \mathbb{R}^{d\times d}$, which determines the extent to which the estimator relies on the map-product dataset. The tuned PTD estimate is defined as

$$\hat{\theta}^{\text{PTD},\hat{\Omega}} = \hat{\Omega}\hat{\gamma}_N + \left(\hat{\theta}_n - \hat{\Omega}\hat{\gamma}_N\right).$$

Algorithm: Predict-Then-Debias Bootstrap

Input: full data set $\mathcal{D} \in \mathbb{R}^{(N+n)\times p}$ parameter estimate function $\hat{\theta}(D)$, bootstrap samples B, error level $\alpha \in (0, 1)$

```
i.) For $b = 1, \ldots, B$:
    (a) Sample $i_1, \ldots, i_{N+n} \overset{\text{iid}}{\sim} \text{Unif}(\{1, \ldots, N+n\})$
```

(b) Let $\mathcal{I} = \{k \in \{1, \dots, N+n\} : D_i \in D$

(c) Let $\tilde{\mathcal{I}} = \{i \in \{1, \dots, N+n\} : D_i \in \tilde{D}$

(d) $\hat{\theta}_n^{(b)} \leftarrow \hat{\theta}_n \left(\{(D_i^{\text{easy}}, D_i^{\text{hard}})\}_{i \in \mathcal{I}} \right)$

(e) $\hat{\gamma}_n^{(b)} \leftarrow \hat{\theta}_N \left(\{(D_i^{\text{easy}}, \tilde{D}_i^{\text{hard}})\}_{i \in \tilde{\mathcal{I}}} \right)$

(f) $\hat{\gamma}_N^{(b)} \leftarrow \hat{\theta}_N \left(\{(D_i^{\text{easy}}, \tilde{D}_i^{\text{hard}})\}_{i \in \tilde{\mathcal{I}}} \right)$

ii.) Select tuning matrix $\hat{\Omega}$

(a) $\hat{\Sigma}_{\theta,\gamma,n} \leftarrow (N+n)\hat{\text{Cov}}(\{\hat{\theta}_n^{(b)}\}_{b=1}^B, \{\hat{\gamma}_n^{(b)}\}_{b=1}^B)$

(b) $\hat{\Sigma}_{\gamma,n} \leftarrow (N+n)\hat{\text{Var}}(\{\hat{\gamma}_n^{(b)}\}_{b=1}^B)$

(c) $\hat{\Sigma}_{\gamma,N} \leftarrow (N+n)\hat{\text{Var}}(\{\hat{\gamma}_N^{(b)}\}_{b=1}^B)$

(d) $\hat{\Omega} \leftarrow \hat{\Sigma}_{\theta,\gamma,n}(\hat{\Sigma}_{\gamma,n} + \hat{\Sigma}_{\gamma,N})^{-1}$

iii.) Compute debiased bootstrap estimates:

$$\hat{\theta}^{\text{PTD},\hat{\Omega},(b)} = \hat{\Omega}\hat{\gamma}_N^{(b)} + \left(\hat{\theta}_n^{(b)} - \hat{\Omega}\hat{\gamma}_N^{(b)}\right), \quad b = 1, \dots, B$$

iv.) Construct confidence intervals using empirical quantiles:

$$C_j^{1-\alpha} = \left(\hat{Q}_{\alpha/2}\left(\{\hat{\theta}_j^{\text{PTD},\hat{\Omega},(b)}\}_{b=1}^B \right), \ \hat{Q}_{1-\alpha/2}\left(\{\hat{\theta}_j^{\text{PTD},\hat{\Omega},(b)}\}_{b=1}^B \right) \right), \quad \forall j \in \{1, \dots, p\}.$$

Output: $(1-\alpha)$-level confidence intervals $C_j^{1-\alpha}$ for each j.

Then under the assumptions that (i) all observations are IID, $D_{i=1}^{N+n} \overset{\text{i.i.d.}}{\sim} \mathbb{P}$, and (ii) the sampling of gold-standard data is done randomly (*missing at random*), we have the following convergence result:

$$\sqrt{N}(\hat{\theta}^{\text{PTD},\hat{\Omega}} - \theta) \xrightarrow{d} \mathcal{N}(0, \Sigma^{\text{PTD}}(\Omega)) \text{ as } N \to \infty,$$

where $\Sigma^{\text{PTD}}(\Omega)$ is defined as:

$$\Sigma^{\text{PTD}}(\Omega) := \Sigma_{\theta,n} - \Sigma_{\theta,\gamma,n}\Omega^{\intercal} - \Omega\Sigma_{\theta,\gamma,n}^{\intercal} + \Omega(\Sigma_{\gamma,n} + \Sigma_{\gamma,N})\Omega^{\intercal}$$

2.4.3 PTD Extensions

The PTD bootstrap algorithm above is the simplest of six provided by Kluger et al. (2025), and assumes the data is independently and identically sampled from the population of interest. This covers many use-cases, however there is great reason to sample differently. For this purpose, two altered bootstrap algorithms are highly relevant; the first for cluster sampling, and the second for stratified sampling. These require additional math and weak assumptions about sampling.

Chapter 3

Data Simulation

3.1 Goal

To evaluate the effectiveness of PPI++ and PTD, two experimental settings are considered. In both, data is first generated from a deterministic process (*i.e.* $Y = \beta X$). Ground truth observations are produced by adding Gaussian noise to this deterministic signal. Four additional map products are simulated by introducing further noise and possible biases to each X, Y, and both X and Y.

3.2 Simulation Outline

For each setting, n is looped over the 6 specified values. For each value of n, the following procedure is applied:

i.) Loop over each map product: MP1, MP2, MP3, annd MP4.

ii.) For each product, compute parameter estimates using:

- *gt*: OLS with gold-standard data only
- *naive*: OLS using map-product data
- *ppi*: PPI++ estimation using all data
- *ptd*: PTD estimation using all data

iii.) For each method, store:

- Empirical coverage rate over 475 simulations (SE $\approx$ 1%).
- Average confidence interval width over 475 simulations.

3.3 Simple Setting

3.3.1 Data Generation

The first setting models a basic linear relationship between a dependent variable Y and an independent variable X.

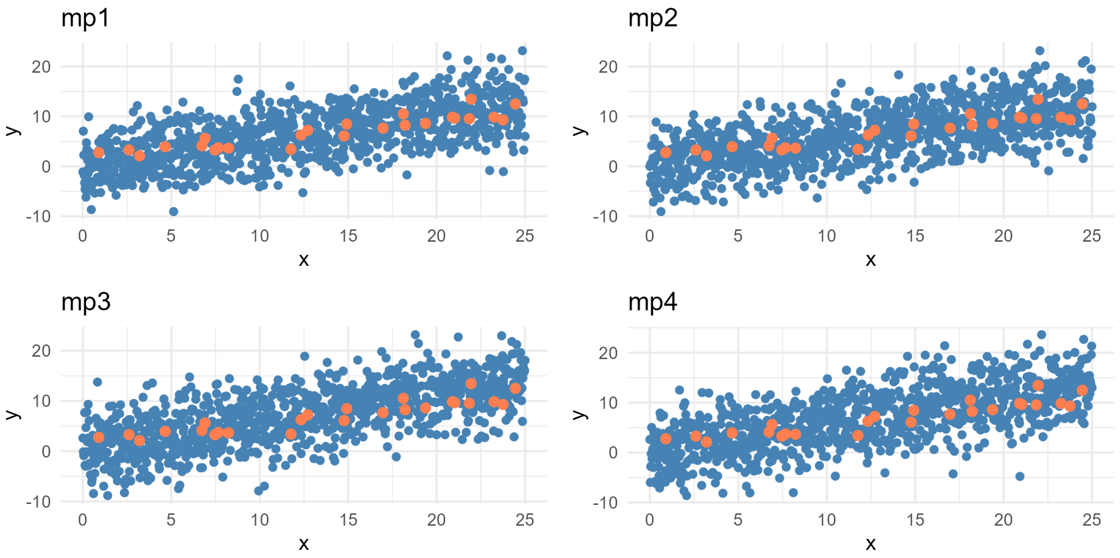

Figure 3.1: Simulated data in the simple setting. Orange points are example gold standard data, while blue points are the biased and noisy mp data

Performance is assessed using 475 simulation runs - to achieve ≈ 1% SE for coverage probability - for 6 different amounts of supplemental ground truth data - $n = (30, 50, 100, 200, 400, 1000)$ and 1000 supplemental noisy data points. For each case, confidence intervals and point estimates are computed using (i) ground truth only, (ii) naive estimation treating the map product as ground truth, (iii) PPI++, and (iv) PTD. These outputs are compared in terms of average confidence interval width and empirical coverage rates.

For each value of X, the gold-standard value of Y is determined by the following:

$$Y = \frac{1}{2}X + \epsilon, \quad \epsilon \sim \mathcal{N}(0, 2)$$

where the gold-standard dataset is chosen a random subset of n points from the 1000. Additionally, the map-product datasets (MP) are generated by adding additional noise and bias as in Table 3.1. The corresponding formulas for map-product data generation can be seen in Table 3.2

Dataset	Bias in X	Bias in Y	Noise (ϵ)
mp1 (extra noise)	-	-	4
mp2 (error in X)	1.1	-	4
mp3 (error in Y)	-	1.1	4
mp4 (error in both)	1.1	0.9	4

Table 3.1: Summary of bias and noise structure for ground truth and MP datasets.

3.3.2 Confidence Interval Coverage

The provided Figure 3.2 is truncated to make viewing easier. For mp2, mp3, and mp4, naive setimation coverage hovers around 0-25% coverage; the full plot can be seen in

Dataset	$\tilde{X}$ formula	$\tilde{Y}$ formula
mp1 (extra noise)	-	$\tilde{Y} = \frac{1}{2}X + \epsilon_{mp1}$
mp2 (error in X)	$\tilde{X} = 1.1X$	$\tilde{Y} = \frac{1}{2}X + \epsilon_{mp2}$
mp3 (error in Y)	-	$\tilde{Y} = \frac{1.1}{2}X + \epsilon_{mp3}$
mp4 (error in both)	$\tilde{X} = 1.1X$	$\tilde{Y} = \frac{0.9}{2}X + \epsilon_{mp4}$

Table 3.2: Formulas for X and Y in ground truth and MP datasets.

Figure A.1. The confidence interval coverage, shown in Figure 3.2, aligns with observations reported by Angelopoulos et al. (2024), except in the case when n is very low ($n \approx 100$ or less), in which we see a small dip. For the extra noise and error-in-Y settings (mp1, mp3), the dip is very minor, and for the cases where an error-in-X is present (mp2, mp4) the dip is more pronounced, which is expected as PPI++ doesn't provide coverage guarantee in this setting. The PTD coverage also looks as expected, except with an exceptionally low amounts of ground truth data ($n \approx 30$). For naive estimation (treating map-product data as gold-standard), the results are great in the first setting, with only additional noise, however is worse in every other setting. We see that coverage is dependent on how biased the data is; if present, coverage will almost certainly be worse than expected (in this case, much worse than 95%).

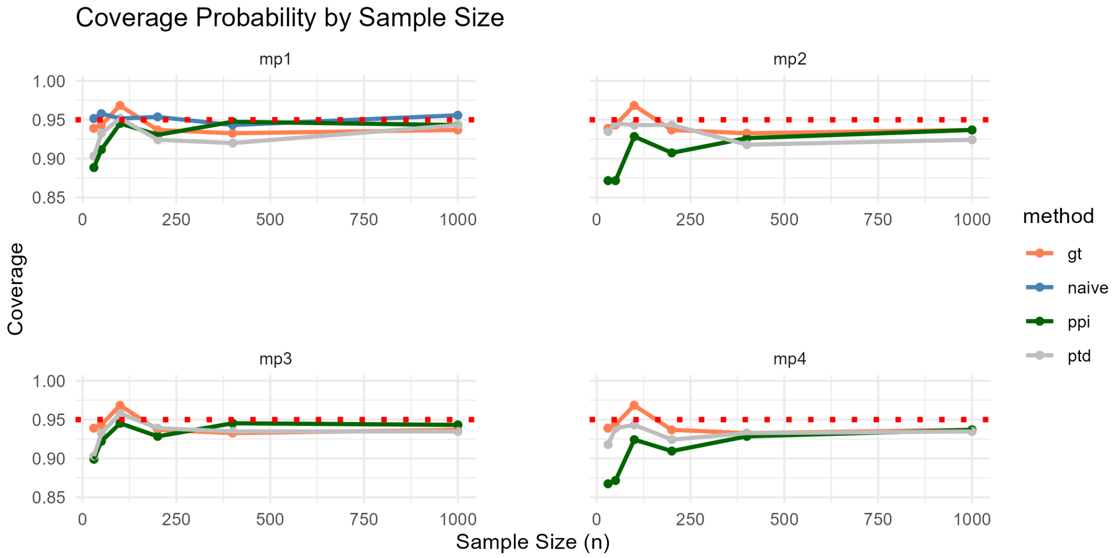

Figure 3.2: Confidence interval coverage in the simple setting

3.3.3 Confidence Interval Width

As shown in Figure 3.3, PPI++ produces noticeably narrower confidence intervals without compromising coverage; this makes it advisable to use when you have enough gold-standard data to avoid dips in coverage. PTD produces smaller confidence intervals, although very marginally so, except in the second setting, MP2. Coverage is very consistent, so it is still potentially reasonable to use PTD when unsure about the quality of map-product data.

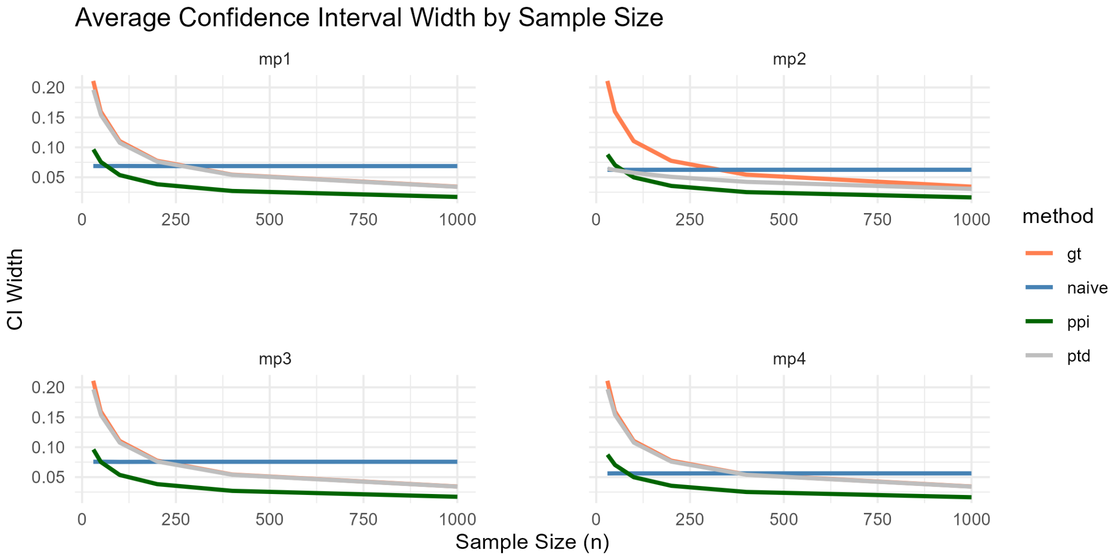

Figure 3.3: Average confidence interval width in the simple setting

3.4 Large Simple Setting

3.4.1 Data Generation

This setting is the same as the simple setting, but tests PPI++ and PTD with larger amounts of data. Performance is still assessed using 475 simulation runs for 6 different amounts of supplemental ground truth data, but now there are n = (100, 200, 500, 1000, 2500, 5000) ground-truth points and 25000 map-product points. These amounts of data are close to the amounts used to test PPI++ and PTD in their respective papers.

3.4.2 Confidence Interval Coverage

The provided figure 3.4 is truncated to make viewing easier. For mp2, mp3, and mp4, naive setimation coverage hovers around 0-25% coverage; the full plot can be seen in Figure A.2. In the setting with more data, the empirical coverage rates (3.4) behave well for each tested amount of gold-standard data. For setting with error-in-X (MP2, MP4), the coverage hovers clearly around 90% instead of 95%, which makes sense, as there's no theoretical guarantees for this setting. For PTD, there is clear 95% coverage as seen in Kluger et al. (2025).

3.4.3 Confidence Interval Width

Confidence interval width (3.5) for all methods matches closely to the simple setting from before. The only notable difference is that in the setting with error-in-X only (MP2), the width is much tighter, overlapping even the naive estimate.

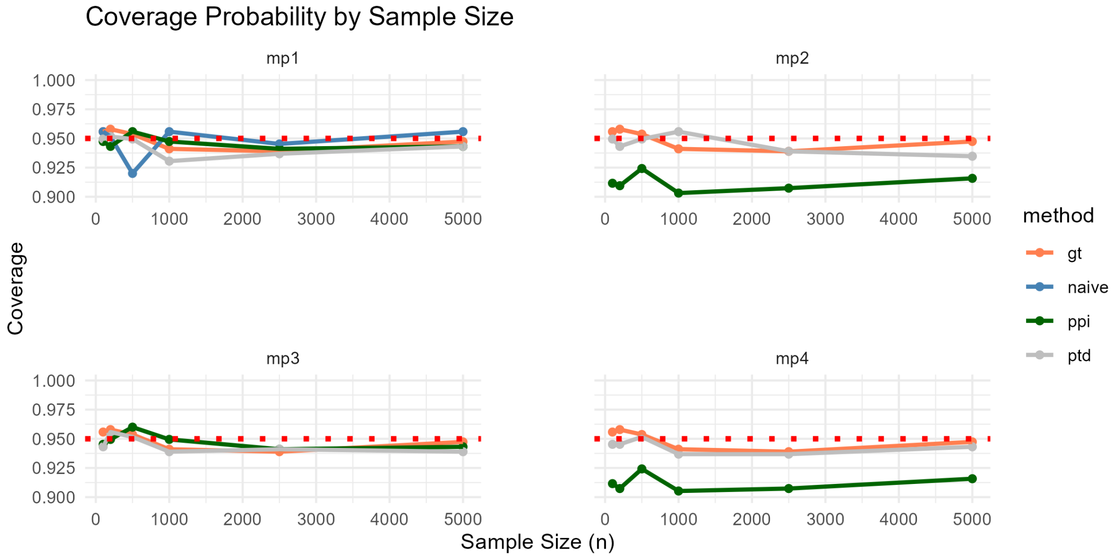

Figure 3.4: Average confidence interval width in the large simple setting

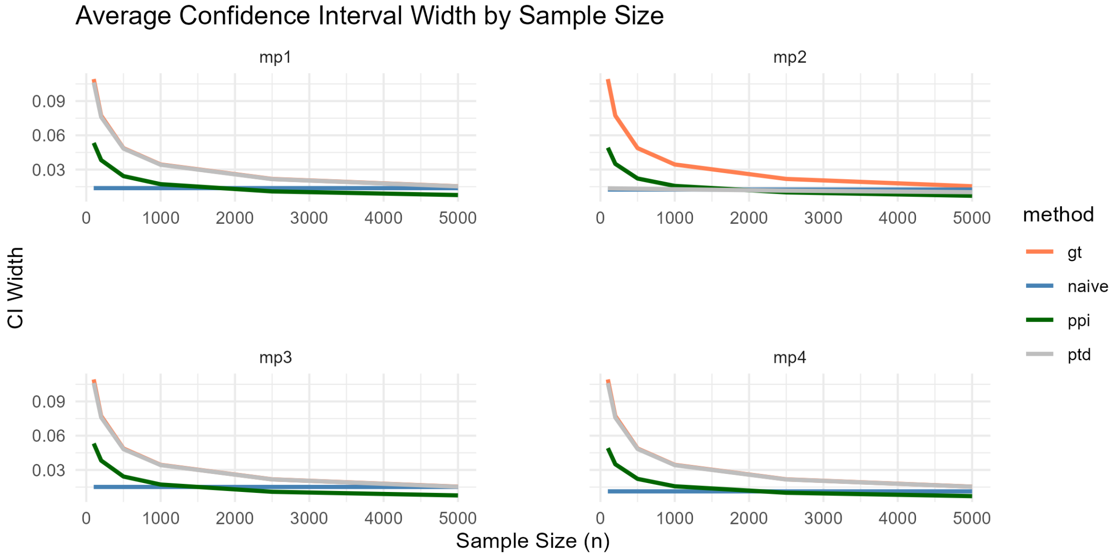

Figure 3.5: Average confidence interval width in the large simple setting

3.5 Two-Dimensional Setting

3.5.1 Data Generation

The second setting simulates spatial data on a 25 × 25 grid of latitude and longitude coordinates. The dependent variable Z is influenced by lat, lon, and a derived variable $dist$, representing distance to the line $y = 8X - 76$. This line emulates a geographic feature such as a river or road (see Figure 3.6).

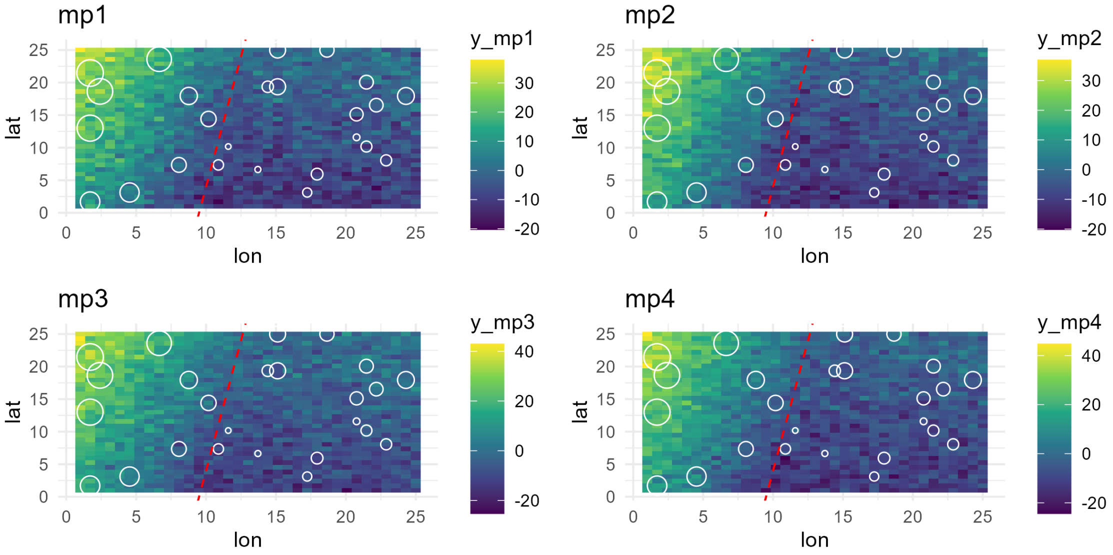

Figure 3.6: Simulated data in the lat-lon setting

Performance is also assessed using 475 simulation runs - to achieve ≈ 1% SE- for 6 different amounts of supplemental ground truth data - $n = (70, 100, 200, 400, 1000)$. For each case, confidence intervals and point estimates are computed using (i) ground truth only, (ii) naive estimation treating the map product as ground truth, (iii) PPI++, and (iv) PTD. These outputs are compared in terms of average confidence interval width and empirical coverage rates.

Data generation is based on the following equation:

$$Z = \frac{2}{3}lat - \frac{4}{3}lon + \frac{3}{2}dist + \epsilon, \quad \epsilon \sim \mathcal{N}(0, 2)$$

with bias and noise for each setting shown in Table 3.3. The corresponding formulas for map-product data generation can be seen in Table 3.4

Dataset	Bias in lon	Bias in lat	Bias in dist	Bias in Y	Noise (σ)
mp1 (extra noise)	-	-	-	-	4
mp2 (error in x)	1.1	-	1.1	-	4
mp3 (error in y)	-	-	-	1.1	4
mp4 (error in both)	1.1	-	0.9	0.9	4

Table 3.3: Summary of bias and noise structure for ground truth and mp datasets.

Dataset	$\tilde{\text{lon}}$ formula	$\tilde{\text{dist}}$ formula	$\tilde{Y}$ formula
mp1 (extra noise)	-	-	$\tilde{Y} = \text{lon} + \text{lat} + \text{dist} + \epsilon_{mp1}$
mp2 (error in x)	$\tilde{\text{lon}} = 1.1\,\text{lon}$	$\tilde{\text{dist}} = 1.1\,\text{dist}$	$\tilde{Y} = \text{lon} + \text{lat} + \text{dist} + \epsilon_{mp2}$
mp3 (error in y)	-	-	$\tilde{Y} = 1.1(\text{lon} + \text{lat} + \text{dist}) + \epsilon_{mp3}$
mp4 (error in both)	$\tilde{\text{lon}} = 1.1\,\text{lon}$	$\tilde{\text{dist}} = 0.9\,\text{dist}$	$\tilde{Y} = 0.9(\text{lon} + \text{lat} + \text{dist}) + \epsilon_{mp4}$

Table 3.4: Formulas for lon, dist, and Y in ground truth and mp datasets. Lat remains unchanged in the inputs.

3.5.2 Confidence Interval Coverage

The provided figure 3.7 is truncated to make viewing easier. For any variable/setting combination where naive estimation coverage is not visible, coverage is near 0%. The full plot can be seen in Figure A.3. Similar overall trends are observed as in the simple setting. Figure 3.7 shows that PPI++ again suffers from a dip in coverage when the proportion of gold-standard data is especially small. The PTD estimator, in contrast, maintains more stable coverage across the range of n, which is consistent with its theoretical guarantees. Naive estimation continues to under-perform, with coverage depending heavily on whether bias is present in the simulated map-products.

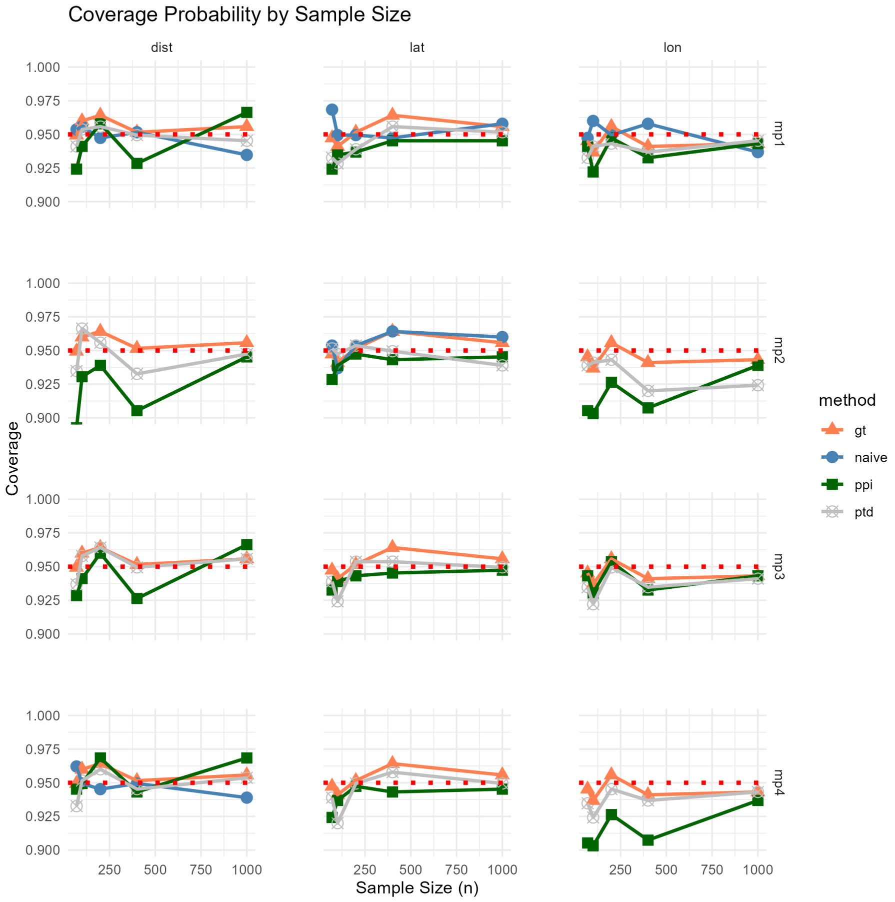

Figure 3.7: Confidence interval coverage in the 2d setting

3.5.3 Confidence Interval Width

Results for confidence interval width, shown in 3.8, also echo the patterns from the simple setting. PPI++ tends to yield narrower intervals than ground truth alone, without sacrificing much coverage once n is moderate. PTD generally produces intervals that are similar in width, but has the advantage of being robust when X as well as Y is corrupted. Taken together, the two-dimensional experiments illustrate that the strengths and limitations observed in the simpler linear setting carry over to higher-dimensional, spatially structured data

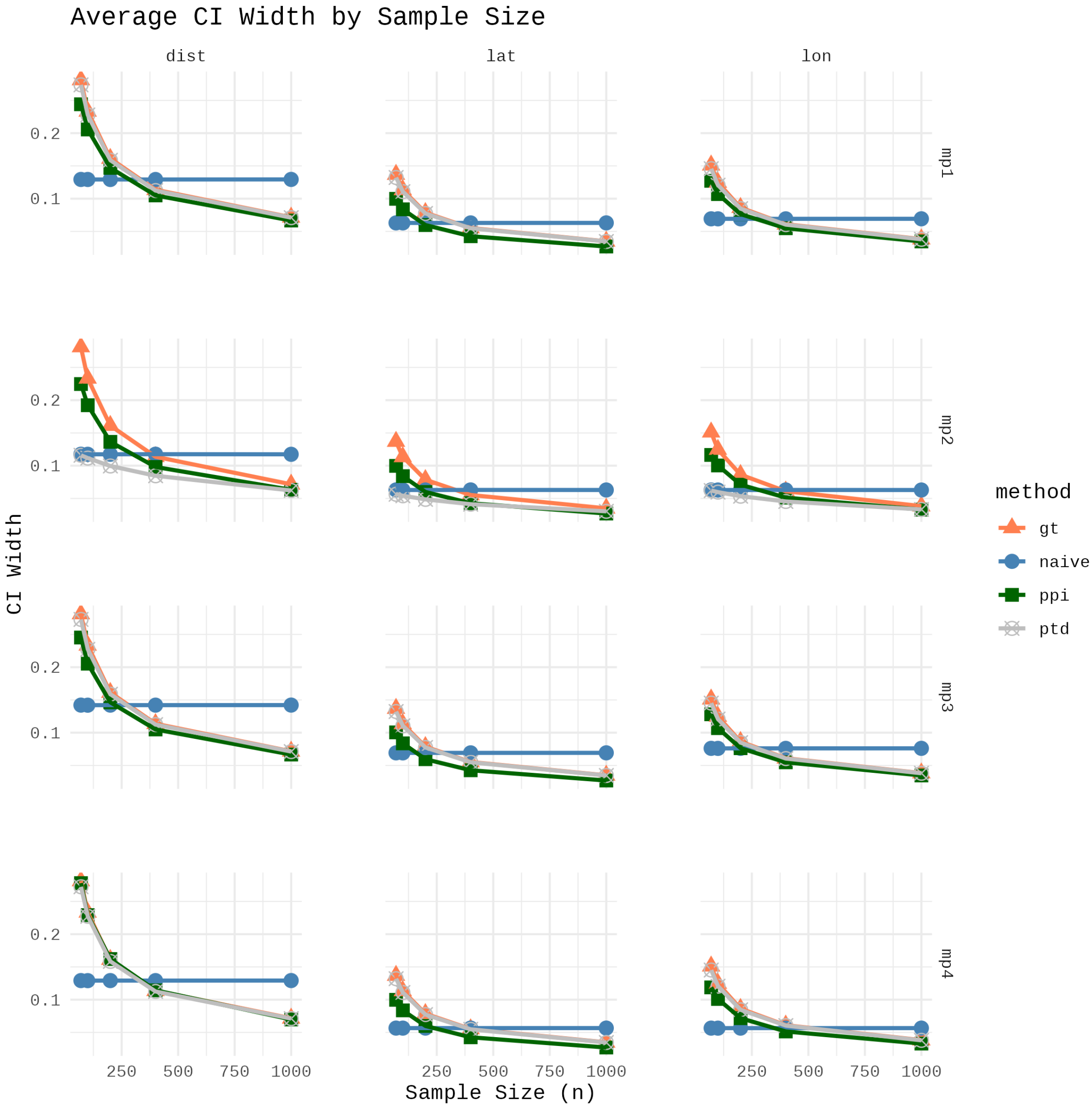

Figure 3.8: Confidence interval width in the 2d setting

3.6 Summary of Simulation Results

The simulations highlight key differences between PPI++ and PTD. When error was limited to the outcome variable Y, PPI++ consistently outperformed PTD, producing narrower confidence intervals while maintaining reasonable coverage. In contrast, PTD was more robust when covariates were also corrupted, though typically less efficient.

A notable finding was the drop in coverage at very small gold-standard sample sizes, especially for PPI++. This effect has not been reported in the original papers, as neither tested such small n, and it shows a practical limitation when only few accurate observations are available.

Overall, PPI++ reproduced the strong performance documented by Angelopoulos et al. (2024), while PTD appeared somewhat weaker than in Kluger et al. (2025). In many settings tested, the gain in confidence interval width are negligible. This may reflect the bootstrap choice or the additional bias structures introduced here.

Chapter 4

Applications

4.1 Areas of Application

4.1.1 When To Consider PPI++

PPI++ can be used when we can obtain X relatively easy, but Y may be expensive to obtain. Coverage results are not theoretically sound when an error-in-X is present, so it is not advisable to use it in that scenario.

4.1.2 When To Consider PTD

PTD can be used when any (not strict) subset of predictors and response are difficult to measure, which also includes any scenario where we could use PPI++. In addition, we can use PTD when sampling of the ground-truth points done randomly, but stratified or clustered, which is particularly useful for epidemiological studies and demanding field surveys.

4.2 Conditions for Application

4.2.1 Conditions for PPI++

Two necessary conditions to apply PPI++ are a pair of datasets (ground-truth and map-product) and a function $f : X \mapsto \hat{Y}$, which is typically a black-box prediction model. Note that we don't necessarily need f if we have $\tilde{Y}_{n+1}, \ldots, \tilde{Y}_{n+N}$ from another source, but in this case it is necessary to also have $\tilde{Y}_1, \ldots, \tilde{Y}_n$ (predictions which accompany the ground-truth data); this is the case for PPI++ example below.

Additionally, we need to be sure that our ground-truth samples of covariates (orange X's in table) are sampled from the same distribution as the map-product samples of covariates (blue X's); this is to avoid possible sampling biases.

Index	X_1	$\dots$	X_p	Y
1	$X_{1,1}$	$\dots$	$X_{1,p}$	Y_1
2	$X_{2,1}$	$\dots$	$X_{2,p}$	Y_2
3	$X_{3,1}$	$\dots$	$X_{3,p}$	Y_3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	$X_{n,1}$	$\dots$	$X_{n,p}$	Y_n
$n+1$	$\tilde{X}_{n+1,1}$	$\dots$	$\tilde{X}_{n+1,p}$	$\tilde{Y}_{n+1}$
$n+2$	$\tilde{X}_{n+2,1}$	$\dots$	$\tilde{X}_{n+2,p}$	$\tilde{Y}_{n+2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n+N$	$\tilde{X}_{n+N,1}$	$\dots$	$\tilde{X}_{n+N,p}$	$\tilde{Y}_{n+N}$

Table 4.1: Data-structure for PPI++. Blue indicates map-product data. Orange is gold-standard

4.2.2 Conditions for PTD

To use PTD in analysis, we need a ground-truth dataset (orange), and map-product dataset (blue). Recall that $\mathcal{D}^{easy}$ is our set of easy to measure variables (which can be any combination of predictors and response), meaning we have ground-truth data for all samples. $\mathcal{D}^{hard}$ is the small sample of ground-truth data for our hard to measure variables. $\tilde{\mathcal{D}}^{hard}$ is the map-product/remotely-sensed data for our hard to measure variables. For example, if we have that Y is hard-to-measure and X is easy to measure, then $\mathcal{D}^{easy} = X$, $\mathcal{D}^{hard} = Y$, and $\tilde{\mathcal{D}}^{hard} = \tilde{Y}$

For PTD, we have a few conditions on our data. First, we need to observe map-product data in the same locations where we observe our ground-truth data. However, in the case where map-product observations of $\{\tilde{\mathcal{D}}^{\text{hard}}\}_{n+1:n+N}$ are obtained via a function f (think PPI++), we can use it to obtain $\{\tilde{\mathcal{D}}^{\text{hard}}\}_{1:n}$. Second, whether or not we see a ground truth observation should not be dependent on the values of the observation (missing at random). Last, we should know how ground-truth data is sampled, and it should be possible for each map-product point to have been sampled as a truly-observed point. The most convenient way to achieve this is to obtain a map-product, then decide randomly some number of points to observe with precision. And although I don't cover an example here, recall that sampling of ground-truth points may also be a stratified random sample, cluster random sample, or some unequal random sample.

Index	$\mathcal{D}^{\text{easy}}$	$\mathcal{D}^{\text{hard}}$	$\tilde{\mathcal{D}}^{\text{hard}}$
1	$\mathcal{D}_1^{\text{easy}}$	$\mathcal{D}_1^{\text{hard}}$	$\tilde{\mathcal{D}}_1^{\text{hard}}$
2	$\mathcal{D}_2^{\text{easy}}$	$\mathcal{D}_2^{\text{hard}}$	$\tilde{\mathcal{D}}_2^{\text{hard}}$
3	$\mathcal{D}_3^{\text{easy}}$	$\mathcal{D}_3^{\text{hard}}$	$\tilde{\mathcal{D}}_3^{\text{hard}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	$\mathcal{D}_n^{\text{easy}}$	$\mathcal{D}_n^{\text{hard}}$	$\tilde{\mathcal{D}}_n^{\text{hard}}$
$n+1$	$\mathcal{D}_{n+1}^{\text{easy}}$		$\tilde{\mathcal{D}}_{n+1}^{\text{hard}}$
$n+2$	$\mathcal{D}_{n+2}^{\text{easy}}$		$\tilde{\mathcal{D}}_{n+2}^{\text{hard}}$
$\vdots$	$\vdots$		$\vdots$
$n+N$	$\mathcal{D}_{n+N}^{\text{easy}}$		$\tilde{\mathcal{D}}_{n+N}^{\text{hard}}$

Table 4.2: Data-structure for PTD. Blue indicates map-product data. Orange is gold-standard

4.3 Example 1: PPI++ Logistic regression

4.3.1 Data Source

This dataset was obtained from the UCI Machine Learning Repository and adapted to illustrate a problem suitable for PPI++. Using only the white wine portion, a subset was selected and each observation was assigned an artificial label of "expert" or "novice." The original quality column was converted to a binary variable, `high_quality`, and the `density` and `chloride` variables were scaled to enhance the interpretability of the plots.

4.3.2 Setting

In this scenario, suppose you and a large group of friends rate wines for fun, either low-quality or high-quality, and also record a bunch of the properties. Over a few years you rate 3000 wines, and you decide that you want to see if the physical and chemical can predict the quality of the wine, but you know the 3000 ratings aren't completely accurate. So your group of friends hires an expert wine-taster to taste 100 of the wines you've also tasted at random.

You could do the analysis on just the 100 expert-rated wines, or naively use all 3100 observations and risk potential bias. In this case, you can also use PPI++ to incorporate both datasets.

As this dataset is constructed and truncated, we can consider parameter obtained from logistic regression on the full dataset as being close-enough proxies for the true parameters of interest. This will be used to compare different methods in this example, as well as the following example.

The form of the data frame used in analysis is shown in Table 4.3. Columns 1-11 are the physical and chemical properties of each wine, while column 12 is a boolean response: 1 if the wine is high quality, and 0 otherwise.

acid	sugar	free sd	total sd	alc	chlor	dens.	sulph	pH	vol acid	citr acid	quality
6.8	11.2	44.0	136.0	9.2	1.1	0.9	3.4	-1.4	0.9	1.1	0
7.5	7.7	61.0	209.0	11.1	0.1	0.0	-1.7	-0.3	-0.5	1.3	0
6.7	4.3	57.0	124.0	10.7	-0.8	-0.8	0.4	-1.8	-1.4	1.4	0
7.9	12.9	13.0	63.0	13.0	-0.6	-0.3	-0.5	-1.3	0.7	-0.4	0
6.3	8.1	44.0	129.0	12.1	-0.4	-0.5	-1.8	0.5	0.0	0.1	1

Table 4.3: Example rows of wine quality dataset

4.3.3 Analysis

```
set.seed(1)

wine ← read_csv("data/wine_quality_labelled.csv")

# separate data into expert and novice dfs
wine_exp ← wine %>%
  filter(label == "expert") %>%
  select(-label)

wine_nov ← wine %>%
  filter(label == "novice") %>%
  select(-label)

# separate the wines tasted by both expert and novices (Y1,
    f(X1)), ..., (Yn, f(Xn))
wine_compare ← wine %>%
  filter(label == "compare") %>%
  select(-label)
```

```
# ppi_plusplus_logistic requires matrices as inputs
# data needs additional intercept column
X_l ← cbind(as.matrix(wine_exp[, -12]))
Y_l ← as.matrix(wine_exp[, 12])
f_l ← as.matrix(wine_compare[, 12])
X_u ← cbind(as.matrix(wine_nov[, -12]))
f_u ← as.matrix(wine_nov[, 12])

# stores all ppi++ outputs and intermediate values
wine_ppi ← ppi_plusplus_logistic(
  X_l = cbind(1, X_l),    # X
  Y_l = Y_l,              # Y
  f_l = f_l,              # f(X)
  X_u = cbind(1, X_u),    # ∼X∼
  f_u = f_u,              # f(∼X∼)
)

# coef and se estimates for ppi++
ppi_coef ← t(wine_ppi$est)[-1]
ppi_se ← wine_ppi$se[-1]
```

4.3.4 Results

Using PPI++, we see smaller confidence intervals than if we just used the ground truth data while maintaining theoretical 95% coverage of true parameters. When comparing PPI++ to just using expert data, we have one additional significant result correctly found, with zero incorrect results added.

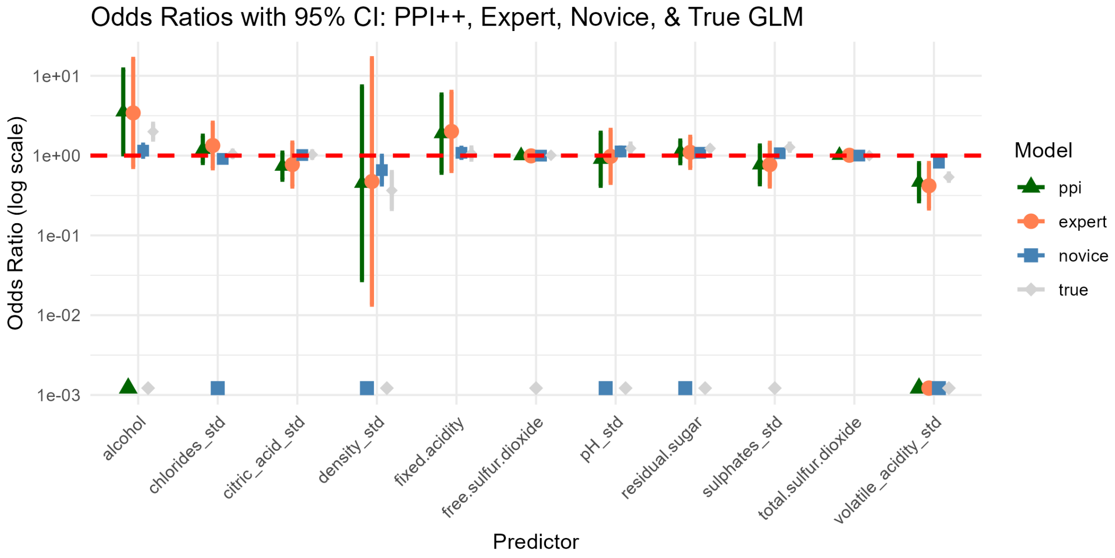

Figure 4.1: Confidence intervals for wine analysis (shape below interval indicates a significant result)

4.4 Example 2: PTD Linear regression

4.4.1 Data Source

The forest cover data set is taken from (paper: Regression coefficient estimation from remote sensing maps). This example uses a different amount of the whole data: n = 500 ground-truth observations, and N = 10000 supplemental observations. There are two predictors, population and elevation, and the goal is determine regression coefficients for their effect on forest cover. Here, forest cover and elevation are map-products, so are calculated from remotely-sensed data. Table 4.4 shows the form of the dataset. And similar to the first example, there is extra gold-standard data that goes unused in analysis, and linear regression on this full dataset acts as a proxy for the true parameters.

	cover	pop	elev
1	48.35	2.85	219.48
2	64.48	2.13	641.03
3	3.11	1.91	265.54
4	75.78	2.30	2064.40
5	84.76	0.80	592.11

Table 4.4: Example rows of forest cover dataset

4.4.2 Analysis

```
set.seed(1)

# ground truth data with matching ~D~_hard values
cover_df_all ← read_csv("data/forest_cover_all.csv")[, -1]

# map-product data
cover_df_mp ← read_csv("data/forest_cover_mp.csv")[, -1] %>%
  rename(
    cover = cover_pred,
    pop = pop_pred,
    elev = elev_truth
  )

# ground truth data: D_easy, D_hard
cover_df_gt ← cover_df_all %>%
  select(ends_with("truth")) %>%
  rename(
    cover = cover_truth,
    pop = pop_truth,
    elev = elev_truth
  )

# ground truth data: D_easy, ~D~_hard
cover_df_compare ← cover_df_all %>%
  select(cover_pred, pop_pred, elev_truth) %>%
  rename(
    cover = cover_pred,
    pop = pop_pred,
    elev = elev_truth
  )

# ALL ground truth points (including those excluded from
    example); proxy for true param
cover_df_true ← read_csv("data/forest_cover_true.csv")[, -1] %>%
  rename(
    cover = cover_truth,
    pop = pop_truth,
    elev = elev_truth
  )
```

```
# ptd bootstrapping
cover_ptd ← PTD_bootstrap.glm(
  true_data_completeSamp = cover_df_gt,              #
      ground-truth data
  predicted_data_completeSamp = cover_df_compare,    #
      ground_truth easy with mp hard
  predicted_data_incompleteSamp = cover_df_mp,       # full mp
      data
  regFormula.glm = "cover ~ elev + pop",
  GLM_type = "linear",
  alpha = 0.05,
  B = 2000,                                          # number of
      bootstraps
  TuningScheme = "Optimal",
  speedup = TRUE
)

ptd_coef ← cover_ptd$PTD_estimate[-1]
ptd_lower ← cover_ptd$PTD_Boot_CIs[-1, 1]
ptd_upper ← cover_ptd$PTD_Boot_CIs[-1, 2]
```

4.4.3 Results

For this example, although there isn't a change in the significance results when compared to just using gold-standard data, we see that the PTD estimate is pulled closer to the true value and has a smaller confidence interval, which still contains the true value.

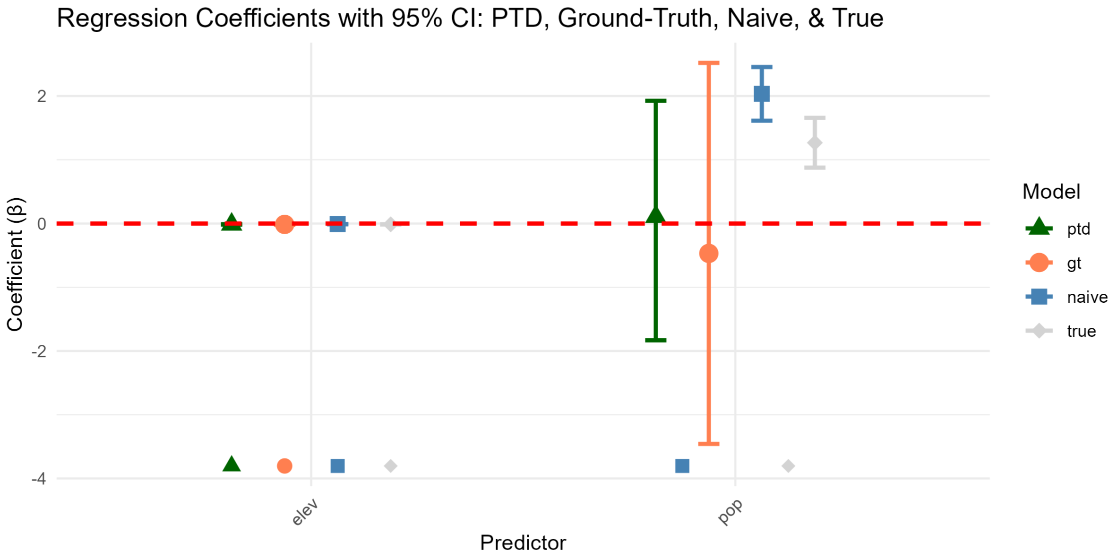

Figure 4.2: Confidence intervals for forest cover analysis (shape below interval indicates a significant result)

4.4.4 Practical Guidelines for Analysis

To summarize the insights from both the simulation studies and applied examples, Table 4.5 provides general guidance on when to prefer PPI++, PTD, or classical inference depending on the data setting.

Data setting	**Method**	**Rationale**
Only Y noisy	PPI++	Narrower CIs without loss of coverage when only the outcome is noisy.
X or X, Y noisy	PTD	Robust to covariate error, maintains stable coverage.
Very small n	Gold-standard only	PPI++ coverage dips at very low n; PTD may also be unstable.
Complex sampling	PTD	Extensions handle stratified/clustered sampling

Table 4.5: Practical guidelines for choosing between PPI++, PTD, and classical inference.

Chapter 5

Conclusion

5.1 Conclusion

This thesis examined prediction-powered inference through both theoretical exposition and empirical evaluation. By studying PPI++, PTD, and their connections to traditional inference, the work highlighted the trade-offs between efficiency and robustness when integrating noisy proxy data with scarce gold-standard labels.

Simulation studies showed that both PPI++ and PTD achieved empirical coverage near 95% under their respective validity assumptions, while also producing narrower confidence intervals than gold-standard-only inference. PPI++ was especially effective when the noisy data only affected the outcome variable, consistently providing efficiency gains. PTD, while slightly less efficient, maintained stable coverage in cases where covariates as well as outcomes were corrupted, underscoring its robustness.

The applied examples reinforced these findings. In the wine quality dataset, PPI++ leveraged proxy labels to detect additional significant predictors without introducing spurious results. In the forest cover dataset, PTD produced estimates closer to the true regression coefficients and tighter intervals compared to using only expert-labeled data. Together, these case studies illustrate that prediction-powered inference methods are not only theoretically sound but also practically beneficial in real-world settings where collecting precise measurements is costly.

In conclusion, PPI++ should be the preferred choice when covariates are reliable and only the response suffers from measurement error, while PTD is the safer option when both predictors and outcomes may be noisy. Both methods represent promising directions for statistical inference in the era of abundant but imperfect data.

5.2 Future Work

This thesis discusses the potential of PPI++ and PTD, but there are still some areas that would be interesting to explore. Further work could clarify how implementation choices, data collection strategies, and bias mechanisms affect their performance. Specifically, the following points are possible extensions to this work:

i.) **Bootstrap tuning in PTD:** The performance of PTD confidence intervals depends on the number of bootstrap samples. Exploring the trade-off between computational cost and coverage accuracy could identify safe defaults or adaptive stopping rules beyond the standard 2000 replicates.

ii.) **Complex sampling designs:** While this work focused on simple random sampling of gold-standard data, PTD extensions allow for stratified and clustered sampling schemes. Creating or finding datasets for these use-cases takes longer as they are highly specific, but testing these extensions empirically would be relevant for epidemiology, survey research, and ecological studies where such sampling is common.

iii.) **Bias structures in simulations:** This thesis considered a single strength of bias applied uniformly across observations. Future simulations could vary both the magnitude and form of bias, including settings where bias occurs *not at random*. Such scenarios may reveal new insights into the robustness and limitations of PPI++ and PTD.

Bibliography

Angelopoulos, A. N., S. Bates, C. Fannjiang, M. I. Jordan, and T. Zrnic (2023). Prediction-powered inference.

Angelopoulos, A. N., J. C. Duchi, and T. Zrnic (2024). Ppi++: Efficient prediction-powered inference.

Bright, B. C., A. T. Hudak, J. M. Egan, C. L. Jorgensen, F. E. Rex, J. A. Hicke, and A. J. Meddens (2020, May). Using satellite imagery to evaluate bark beetle-caused tree mortality reported in aerial surveys in a mixed conifer forest in northern idaho, usa. *Forests 11*(5), 529.

Chen, Y.-H. and H. Chen (2000). A unified approach to regression analysis under double-sampling designs. *Journal of the Royal Statistical Society. Series B (Statistical Methodology) 62*(3), 449–460.

Einbinder, B.-S., L. Ringel, and Y. Romano (2025). Semi-supervised risk control via prediction-powered inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–12.

Kluger, D. M., K. Lu, T. Zrnic, S. Wang, and S. Bates (2025). Prediction-powered inference with imputed covariates and nonuniform sampling.

Lu, K., D. M. Kluger, S. Bates, and S. Wang (2025). Regression coefficient estimation from remote sensing maps.

Poulet, P.-E., M. Tran, S. T. du Montcel, B. Dubois, S. Durrleman, B. Jedynak, and the Alzheimers Disease Neuroimaging Initiative (2025). Prediction-powered inference for clinical trials: application to linear covariate adjustment. *medRxiv*.

Zhang, A., L. D. Brown, and T. T. Cai (2019). Semi-supervised inference: General theory and estimation of means. *The Annals of Statistics 47*(5), 2538 – 2566.

Appendix A

Appendix

A.1 Full Coverage Plot for Simple Setting

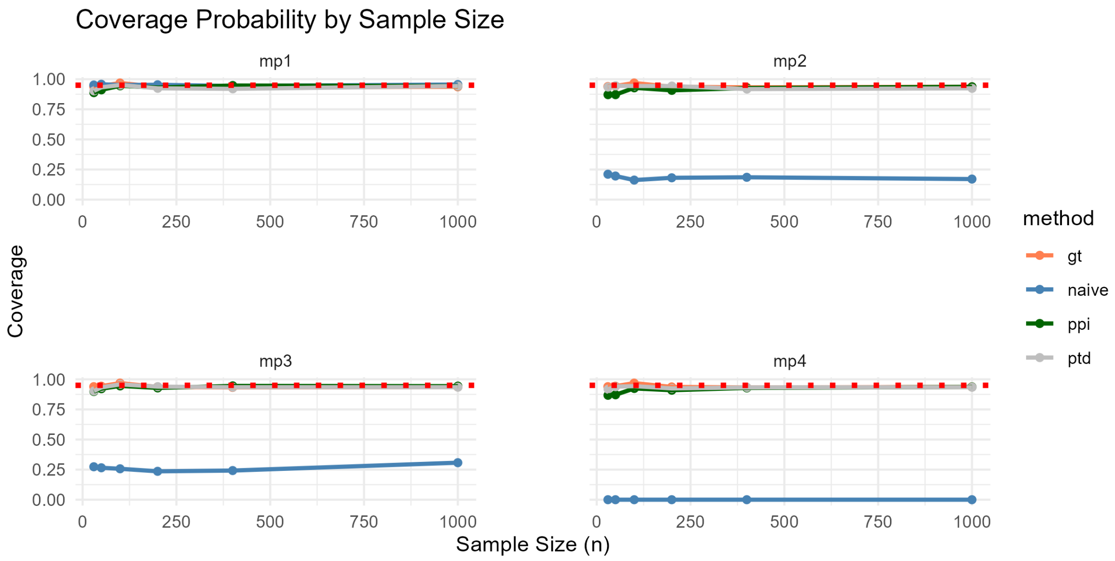

Figure A.1: Confidence interval coverage in the simple setting

A.2 Full Coverage Plot for Large Setting

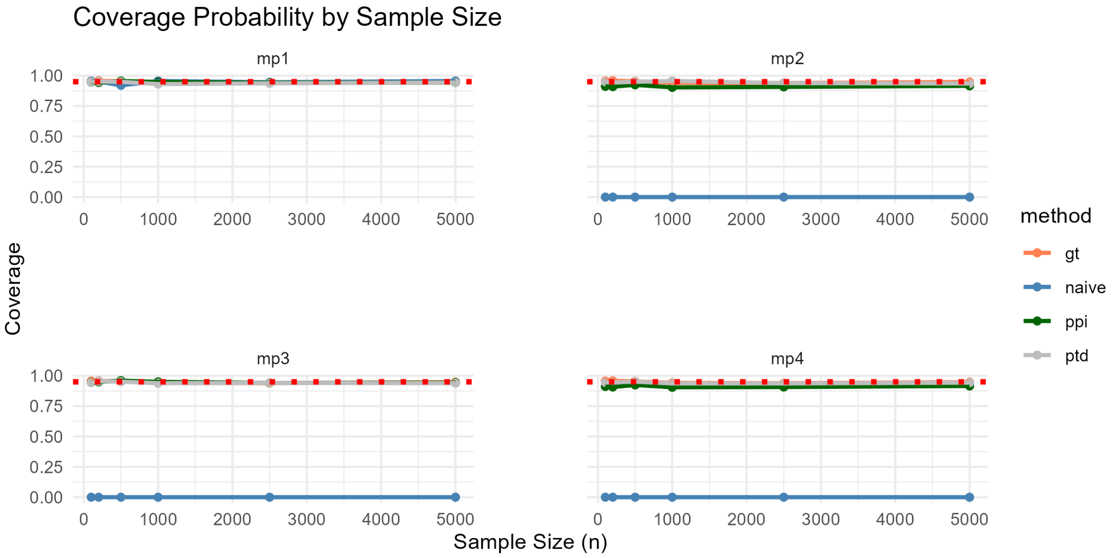

Figure A.2: Average confidence interval width in the large simple setting

A.3 Full Coverage Plot for 2D Setting

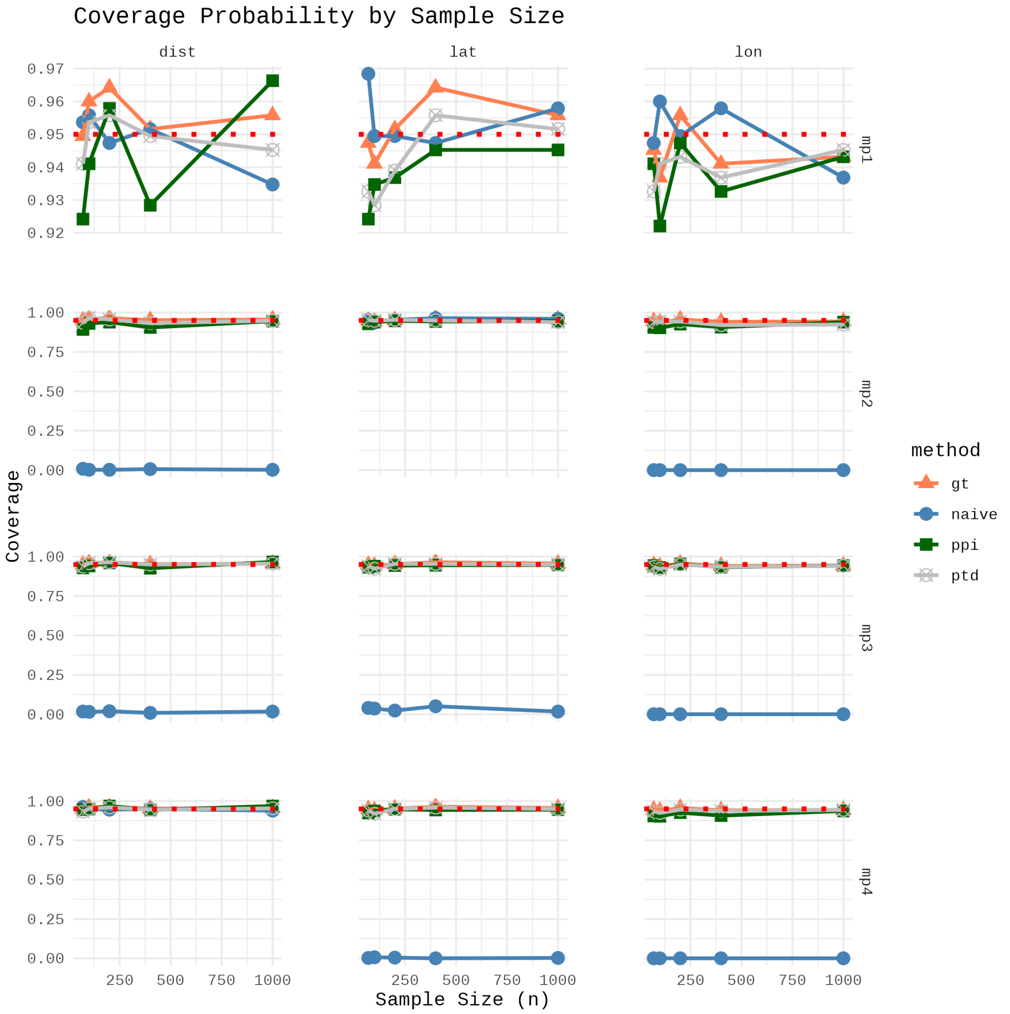

Figure A.3: Confidence interval coverage in the 2d setting

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every written paper or thesis authored during the course of studies. **In consultation with the supervisor**, one of the following two options must be selected:

□ I hereby declare that I authored the work in question independently, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used no generative artificial intelligence technologies[1].

☒ I hereby declare that I authored the work in question independently. In doing so I only used the authorised aids, which included suggestions from the supervisor regarding language and content and generative artificial intelligence technologies. The use of the latter and the respective source declarations proceeded in consultation with the supervisor.

Title of paper or thesis:

Statistical Inference Using Prediction Powered
Inference and Predict-Then-Debias

Authored by:
If the work was compiled in a group, the names of all authors are required.

Last name(s):	**First name(s):**
Li	Roger

With my signature I confirm the following:

- I have adhered to the rules set out in the Citation Guidelines.
- I have documented all methods, data and processes truthfully and fully.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for originality.

Place, date	**Signature(s)**
Zurich, 08.09.2025	

If the work was compiled in a group, the names of all authors are required. Through their signatures they vouch jointly for the entire content of the written work.

[1] For further information please consult the ETH Zurich websites, e.g. https://ethz.ch/en/the-eth-zurich/education/ai-in-education.html and https://library.ethz.ch/en/researching-and-publishing/scientific-writing-at-eth-zurich.html (subject to change).