Does the Policy Actually Work?

Causal Inference for Indonesian Social Protection Programs

Multi-Method Benchmarking with Ground Truth Evaluation

Comparing 13 State-of-the-Art Methods Across Three Paradigms:

Classical Econometrics • Modern Machine Learning • Deep Learning

Applied to PKH Cash Transfers, JKN Healthcare, and Provincial Wage Policies

---

By Hilmi

Contents

Chapter 1

The Business Problem: Why Causal Inference Matters for Indonesia

1.1 The Trillion-Rupiah Question

Indonesia spends over **Rp 450 trillion** (approximately USD 29 billion) annually on social protection programs. These programs reach hundreds of millions of citizens from conditional cash transfers to universal healthcare. But a fundamental question haunts every policymaker, every budget committee, and every development economist:

"Do these programs actually work? And for whom do they work best?"

This is not merely an academic curiosity. Getting the answer wrong has real consequences. If a cash transfer program does not actually improve household welfare, then hundreds of trillions of rupiah are being spent without impact. If a healthcare program works brilliantly for urban populations but fails rural communities, then policy design needs urgent correction. If minimum wage increases reduce employment rather than alleviating poverty, then well-intentioned policies may harm the people they aim to help.

The challenge is that answering these questions is fundamentally harder than it appears. We cannot simply compare people who received a program to those who did not, because the two groups differ in ways that confound the comparison. Families enrolled in cash transfer programs are, by design, poorer, so comparing their outcomes to non-enrolled families tells us about poverty, not about the program's effect.

This is where **causal inference** enters the picture.

1.2 Three Programs, Three Questions

This research focuses on three flagship Indonesian social protection programs, each posing a distinct causal question:

1.2.1 Program Keluarga Harapan (PKH) — Conditional Cash Transfers

PKH is Indonesia's primary conditional cash transfer (CCT) program, reaching approximately 10 million households. Beneficiary families receive cash payments conditional on children's school attendance and family health check-ups. The program uses a *Proxy*

Means Test (PMT) targeting mechanism (Alatas et al., 2012) to identify eligible households based on observable socioeconomic indicators.

Causal Question: What is the effect of PKH enrollment on household welfare? Does this effect vary across household characteristics, for instance, do female-headed households benefit more than male-headed ones? Do rural households benefit differently from urban ones?

Business Stakes: PKH costs approximately Rp 29.1 trillion annually. Understanding heterogeneous treatment effects can guide targeting refinements that maximize welfare impact per rupiah spent.

1.2.2 Jaminan Kesehatan Nasional (JKN) — Universal Health Coverage

Launched in 2014, JKN is one of the world's largest single-payer health insurance systems, covering over 200 million Indonesians (Pisani et al., 2017). The program aims to eliminate financial barriers to healthcare, particularly for maternal and neonatal services.

Causal Question: Does JKN enrollment improve neonatal health outcomes? Among which maternal subgroups, anemic mothers, rural mothers, first-time mothers, is the effect largest?

Business Stakes: Neonatal mortality remains a critical challenge in Indonesia. Understanding which mothers benefit most from JKN can inform targeted interventions that save lives and reduce the economic burden of preventable infant health complications.

1.2.3 Provincial Minimum Wage Policies

Indonesia's decentralized system allows provincial governors to set minimum wages above the national floor. Several provinces have pursued aggressive minimum wage increases, while others have maintained more conservative trajectories (Suryahadi et al., 2003; Del Carpio et al., 2015).

Causal Question: Do aggressive minimum wage increases reduce formal sector employment? What is the magnitude of the employment effect, and does it vary by province characteristics?

Business Stakes: The minimum wage debate directly affects millions of workers and businesses. Evidence-based policy requires rigorous methods that account for province-specific trends and spillover effects.

1.3 Why Simple Comparisons Fail

Consider a naïve analysis: compare average welfare for PKH recipients versus non-recipients. The result would show that PKH recipients have *lower* welfare, not because the program harms them, but because the program *targets* the poorest households. This is **selection bias**, the central challenge of causal inference.

Three forms of bias plague observational studies of social programs:

1. **Selection Bias**: Treated and control groups differ systematically in pre-treatment characteristics. PKH targets poor households; JKN enrollment correlates with health awareness; provinces raising minimum wages differ economically from those that do not.

2. **Confounding**: Unobserved variables jointly influence both treatment assignment and outcomes. Geographic factors, political preferences, or cultural norms may drive both program participation and outcomes.

3. **Heterogeneity**: The effect of a program is rarely uniform. It varies across individuals and contexts. A program that works "on average" may fail entirely for certain subgroups, or may work spectacularly for a subgroup that current targeting misses.

This project deploys 13 different causal inference methods, spanning classical statistics, modern machine learning, and deep learning, to address these challenges rigorously across all three programs.

1.4 What This Project Delivers

By the end of this guide and its accompanying codebase, you will be able to:

1. Understand the fundamental problem of causal inference from first principles.
2. Apply 13 state-of-the-art methods to real policy evaluation problems.
3. Evaluate which methods work best under different conditions using rigorous metrics.
4. Interpret heterogeneous treatment effects for policy-relevant subgroups.
5. Conduct Monte Carlo simulation studies for method comparisons.
6. Make informed recommendations about method selection for specific research questions.

Who Should Read This Guide?

Students: Start with Chapters 1–3 for conceptual foundations, then follow Chapters 4–7 to understand and run every method. **Recruiters**: Chapters 1 and 8–10 demonstrate research scope, technical depth, and honest scientific methodology. **Researchers**: Chapters 4–6 provide complete theoretical and implementation details; Chapter 11 suggests extension directions.

Chapter 2

Foundations of Causal Inference

2.1 The Fundamental Problem

Causal inference seeks to answer: *"What would have happened if things had been different?"* This deceptively simple question leads to one of the deepest challenges in statistics.

2.1.1 The Potential Outcomes Framework

We adopt the **Rubin Causal Model** (also called the potential outcomes framework), formalized by Rubin (1974) and extended by Rosenbaum and Rubin (1983). For each individual i:

- $Y_i(1)$: the *potential outcome* if individual i receives treatment
- $Y_i(0)$: the *potential outcome* if individual i does not receive treatment
- $T_i \in \{0, 1\}$: the observed treatment assignment
- X_i: a vector of pre-treatment covariates

The **Individual Treatment Effect** (ITE) for person i is:

$$\tau_i = Y_i(1) - Y_i(0) \tag{2.1}$$

The **fundamental problem of causal inference** (Holland, 1986) is that we can never observe both $Y_i(1)$ and $Y_i(0)$ for the same individual. Each person is either treated or not treated, we observe one potential outcome and the other remains forever *counterfactual*.

2.1.2 Key Estimands

Since individual effects are unobservable, we target population-level summaries:

Average Treatment Effect (ATE):

$$\text{ATE} = \mathbb{E}[Y(1) - Y(0)] = \mathbb{E}[\tau_i] \tag{2.2}$$

Average Treatment Effect on the Treated (ATT):

$$\text{ATT} = \mathbb{E}[Y(1) - Y(0) \mid T = 1] \tag{2.3}$$

Conditional Average Treatment Effect (CATE):

$$\tau(x) = \mathbb{E}[Y(1) - Y(0) \mid X = x] \tag{2.4}$$

The CATE is particularly important for policy because it reveals *heterogeneity*, how the treatment effect varies across subgroups. A policymaker who knows that PKH benefits rural female-headed households three times more than urban male-headed households can design targeting strategies accordingly.

2.1.3 Identification Assumptions

Estimating causal effects from observational data requires assumptions:

Assumption 1 — Unconfoundedness (Ignorability):

$$(Y(0), Y(1)) \perp T \mid X \tag{2.5}$$

Conditional on observed covariates X, treatment assignment is independent of potential outcomes. Intuitively: if we control for all relevant variables, the remaining variation in treatment is "as good as random."

Assumption 2 — Overlap (Positivity):

$$0 < P(T = 1 \mid X = x) < 1 \quad \text{for all } x \tag{2.6}$$

Every individual must have a positive probability of being either treated or untreated. This ensures we can find comparable individuals across treatment groups.

Assumption 3 — Stable Unit Treatment Value Assumption (SUTVA): Each individual's outcome depends only on their own treatment, not on the treatment status of others. In the context of PKH, this means one household's enrollment does not affect another household's welfare.

Note: When Assumptions Fail

These assumptions cannot be tested directly from data. If unconfoundedness fails (there are important unmeasured confounders), all methods in this study may produce biased estimates. This is a fundamental limitation of observational causal inference. Chapter 12 discusses this further. For the programs in this study, rich covariate sets including geographic, demographic, economic, and health indicators provide strong grounds for approximate unconfoundedness, particularly given the administrative targeting mechanisms used by PKH and JKN.

2.2 The Propensity Score

The **propensity score** (Rosenbaum and Rubin, 1983) is defined as:

$$e(x) = P(T = 1 \mid X = x) \tag{2.7}$$

A landmark result shows that if unconfoundedness holds given X, it also holds given the scalar propensity score $e(X)$. This dramatically simplifies the problem: instead of matching on a high-dimensional covariate vector, we can match on a single number.

The propensity score serves multiple roles across the methods in this project:

- **Matching**: Pairing treated and control individuals with similar propensity scores (Chapter 4).
- **Weighting**: Creating pseudo-populations where treatment is independent of covariates (Chapter 4).
- **Doubly Robust Estimation**: Combining propensity scores with outcome modeling for robustness (Chapter 4).
- **Representation Balancing**: Neural networks learn representations where treatment groups are balanced (Chapter 6).

2.3 From Theory to Practice: The Evaluation Challenge

A central challenge in applied causal inference is: *how do we know our estimates are correct?* In a randomized controlled trial, randomization guarantees unbiased estimates. In observational studies, we have no such guarantee.

This project addresses this challenge through a **benchmarking approach** with known ground truth. For each dataset, we know the true treatment effect at the individual level, enabling us to compute evaluation metrics that would be impossible with purely observational data. This approach follows a rich tradition in the causal inference literature, including the IHDP benchmark (Hill, 2011), the Twins benchmark (Louizos et al., 2017), and the ACIC competition datasets.

Our evaluation metrics include:

$\sqrt{\text{PEHE}}$ (Precision in Estimation of Heterogeneous Effects):

$$\sqrt{\text{PEHE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{\tau}_i - \tau_i)^2} \tag{2.8}$$

This measures how well a method estimates individual-level effects, the gold standard for CATE estimation. Lower is better.

Coverage Rate:

$$\text{Coverage} = \frac{1}{n_{\text{sim}}} \sum_{s=1}^{n_{\text{sim}}} \mathbf{1}\left[\text{ATE}_{\text{true}} \in \text{CI}_s\right] \tag{2.9}$$

The fraction of times the 95% confidence interval contains the true ATE. A well-calibrated method should achieve approximately 95% coverage.

Chapter 3

The Datasets

This chapter describes the three datasets used in our benchmark, each corresponding to one of the Indonesian social protection programs introduced in Chapter 1.

3.1 Dataset 1: PKH Household Survey

3.1.1 Overview

The PKH dataset contains records of **3,000 Indonesian households** across Indonesia's 34 provinces. Each household is characterized by 25 covariates spanning demographics, geography, economics, and social indicators. The binary treatment variable indicates PKH enrollment, and the outcome is a composite household welfare index.

3.1.2 Features

The dataset includes the following covariate groups:

Household Demographics:

- `household_size`: Number of household members (mean ≈ 4.5)
- `n_children`: Number of children under 18
- `head_age`: Age of household head
- `head_female`: Binary indicator for female-headed household
- `head_education_years`: Years of formal education of household head

Geographic Characteristics:

- `is_urban`: Binary indicator for urban location
- `is_java`: Binary indicator for Java island residence
- `province`: Province identifier (1–34)
- `distance_to_facility`: Distance to nearest health/education facility (km)

Economic Indicators:

- `monthly_expenditure`: Monthly household expenditure (Rp, log-transformed)

- `has_savings`: Binary indicator for savings account ownership
- `dwelling_score`: Housing quality composite (0–10)
- `asset_index`: Household asset ownership index
- `land_ownership`: Agricultural land area (hectares)

Employment & Social:

- `employment_formal`: Binary indicator for formal sector employment
- `has_health_insurance`: Binary indicator for any health insurance
- `community_participation`: Community group participation score

Health & Education:

- `child_school_attendance`: School attendance rate for children
- `child_vaccination`: Vaccination completion rate
- `maternal_health_visits`: Number of prenatal check-ups
- `nutrition_score`: Household nutrition adequacy index

Poverty Targeting:

- `pmt_score`: Proxy Means Test score used for PKH targeting
- `poverty_gap`: Distance below poverty line (0 if above)
- `vulnerability_index`: Composite vulnerability measure

3.1.3 Treatment Assignment

Treatment (PKH enrollment) was assigned based on a propensity mechanism reflecting Indonesia's actual targeting process (Alatas et al., 2012). Households with lower PMT scores, rural location, larger household sizes, female heads, and higher vulnerability indices had higher enrollment probabilities. The overall treatment rate is approximately **35.7%**.

3.1.4 Outcome Variable

The `welfare_index` is a composite measure reflecting household well-being across consumption, health, education, and asset dimensions. Higher values indicate better welfare. The ground truth average treatment effect is approximately **5.54 index points**.

Treatment effects are heterogeneous by design: poorer households, rural households, and female-headed households experience larger treatment effects, consistent with findings from rigorous evaluations of CCT programs (Cahyadi et al., 2020).

3.1.5 Benchmark Analog

The PKH dataset's structure mirrors the **IHDP** (Infant Health and Development Program) benchmark (Hill, 2011), widely used in the causal inference literature. Like IHDP, it features moderate sample size, moderate treatment rate, and heterogeneous effects driven by observable covariates.

Note: Data Files

The PKH data is stored in two CSV files: `data/pkh_observed.csv` (observable data with treatment and outcome) and `data/pkh_ground_truth.csv` (individual-level true effects for evaluation). In practice, only the observed file would be available to researchers; the ground truth file enables our benchmarking methodology.

3.2 Dataset 2: JKN Maternal–Neonatal Health Survey

3.2.1 Overview

The JKN dataset contains records of **10,000 mothers** organized in **5,000 matched pairs**. Each mother is characterized by 30 covariates covering maternal demographics, health indicators, prenatal care, delivery characteristics, and socioeconomic factors. The binary treatment indicates JKN enrollment, and the outcome is a neonatal health composite score.

3.2.2 Features

Maternal Demographics:

- `mother_age`: Age at delivery
- `parity`: Number of previous pregnancies
- `education_years`: Years of maternal education
- `is_married`: Marital status
- `is_working`: Employment status during pregnancy

Maternal Health:

- `bmi_prepregnancy`: Pre-pregnancy BMI
- `hemoglobin`: Hemoglobin level (g/dL)
- `is_anemic`: Binary anemia indicator (hemoglobin < 11 g/dL)

- `blood_pressure_systolic`, `blood_pressure_diastolic`: Blood pressure readings
- `has_gestational_diabetes`: GDM indicator
- `has_preeclampsia`: Pre-eclampsia indicator

Prenatal Care:

- `anc_visits`: Number of antenatal care visits
- `first_anc_trimester`: Whether first ANC visit was in first trimester
- `iron_supplement`: Iron supplementation compliance
- `tetanus_vaccination`: Tetanus toxoid vaccination status

Delivery Characteristics:

- `gestational_age_weeks`: Gestational age at delivery
- `birth_weight_grams`: Newborn birth weight
- `delivery_facility`: Type of delivery facility (0=home, 1=clinic, 2=hospital)
- `skilled_birth_attendant`: Whether a skilled provider attended delivery
- `cesarean_delivery`: C-section indicator

Socioeconomic Context:

- `household_expenditure`: Monthly household expenditure
- `is_urban`: Urban residence indicator
- `province`: Province identifier
- `wealth_quintile`: Household wealth quintile (1–5)

3.2.3 Treatment and Outcome

Treatment indicates JKN enrollment. The treatment rate is approximately **69.3%**, reflecting JKN's broad coverage. The outcome variable `neonatal_health_score` is a composite (0–100 scale) reflecting birth weight adequacy, APGAR scores, and absence of neonatal complications.

The true ATE is approximately **8.12 points**. Treatment effects are heterogeneous: anemic mothers, rural mothers, and those with low birth weight infants experience larger benefits, reflecting JKN's role in facilitating access to care that would otherwise be financially prohibitive (Sparrow et al., 2013).

3.2.4 Paired Structure

The JKN dataset has a **paired structure**: each treated mother is matched with a control mother sharing similar baseline characteristics. This structure mirrors the **Twins** benchmark (Louizos et al., 2017), providing an additional layer for counterfactual reasoning.

Note: Data Files

Files: `data/jkn_observed.csv` and `data/jkn_ground_truth.csv`. The `pair_id` column links matched pairs.

3.3 Dataset 3: Provincial Minimum Wage Panel

3.3.1 Overview

The provincial dataset is a **panel** covering all **34 Indonesian provinces** observed over **15 years** (2010–2024), yielding 510 observations. This longitudinal structure enables policy evaluation methods that exploit time variation.

3.3.2 Panel Structure

Each province-year observation includes:

- `province`: Province name
- `year`: Calendar year
- `population`: Provincial population
- `gdp_per_capita`: GDP per capita (million Rp)
- `unemployment_rate`: Unemployment rate (%)
- `min_wage_level`: Nominal minimum wage (million Rp/month)
- `industrial_share`: Manufacturing sector share of GDP (%)
- `education_index`: Provincial education quality index
- `infrastructure_index`: Infrastructure development index

3.3.3 Treatment Definition

Six provinces implemented **aggressive minimum wage increases** at staggered times between 2016 and 2020. The treated provinces are: *Banten*, *Sulawesi Barat*, *Maluku*, *Bangka Belitung*, *Jawa Tengah*, and *Riau*. The staggered adoption creates natural variation in treatment timing that enriches the analysis.

3.3.4 Outcome Variables

The primary outcome is `employment_rate` (formal sector employment rate, %). Secondary outcomes include `formal_sector_share`, `avg_wage`, `gdp_growth`, and `poverty_rate`.

The TWFE DiD estimate of the employment effect is approximately -2.95 **percentage points**, suggesting a negative (though policy-debatable) employment impact from aggressive wage increases.

3.3.5 Benchmark Analog

This dataset's structure mirrors the **California Proposition 99** tobacco control study (Abadie et al., 2010), the canonical application of the Synthetic Control Method. Like that study, we have a small number of treated units, a longer pre-treatment period, and outcome variables that can be well-predicted from donor units.

Note: Data Files

Files: `data/provincial_panel.csv` and `data/provincial_treatment_info.csv`. The treatment info file specifies which provinces were treated and when.

Chapter 4

Classical Causal Inference Methods

This chapter covers three foundational methods rooted in the propensity score literature. These methods have decades of theoretical development and remain workhorses in applied economics and epidemiology.

4.1 Propensity Score Matching (PSM)

4.1.1 Intuition

Matching is perhaps the most intuitive approach to causal inference: for each treated individual, find one or more "similar" control individuals and compare their outcomes. The propensity score provides a principled definition of "similar", individuals with the same probability of treatment.

4.1.2 Theory

PSM proceeds in two stages. First, estimate the propensity score:

$$\hat{e}(x_i) = \hat{P}(T_i = 1 \mid X_i = x_i) \tag{4.1}$$

using logistic regression, gradient boosting, or other classifiers.

Second, for each treated unit i, find the k nearest control units in propensity score space:

$$\mathcal{M}(i) = \underset{j:T_j=0}{\operatorname{argmin}} |\hat{e}(x_i) - \hat{e}(x_j)|, \quad |\mathcal{M}(i)| = k \tag{4.2}$$

The matched estimator for the ATT is:

$$\widehat{\mathrm{ATT}}_{\mathrm{PSM}} = \frac{1}{n_1} \sum_{i:T_i=1} \left(Y_i - \frac{1}{k} \sum_{j \in \mathcal{M}(i)} Y_j \right) \tag{4.3}$$

An optional **caliper** restricts matches to pairs within a maximum propensity score distance, discarding treated units without sufficiently close matches.

4.1.3 Balance Diagnostics

A critical step after matching is verifying that covariates are balanced between matched treated and control groups. The **standardized mean difference (SMD)** for covariate

k is:

$$\text{SMD}_k = \frac{|\bar{X}_{k,T=1} - \bar{X}_{k,T=0}|}{\sqrt{(s^2_{k,T=1} + s^2_{k,T=0})/2}} \tag{4.4}$$

where $\bar{X}$ and s^2 denote sample means and variances. An SMD below 0.1 is generally considered acceptable balance (LaLonde, 1986).

4.1.4 Implementation

Our PSM implementation is in `src/methods/classical/psm.py`. It uses scikit-learn's `GradientBoostingClassifier` for propensity score estimation and k-nearest neighbor matching ($k = 3$ by default). Bootstrap resampling provides confidence intervals. The file also computes balance diagnostics (SMD before and after matching) for all covariates.

Note: PSM Limitations

PSM can only match on *observed* covariates. If treatment assignment depends on unobserved factors, matching will not eliminate bias. Additionally, matching discards unmatched units, potentially reducing effective sample size. In our PKH experiment, PSM achieved an ATE of approximately 5.76 with a relative bias of 4.09%, the highest among classical methods, reflecting the difficulty of propensity score estimation in finite samples.

4.2 Inverse Propensity Weighting (IPW)

4.2.1 Intuition

Rather than matching individuals, IPW *reweights* the entire sample to create a pseudo-population in which treatment is independent of covariates. Individuals who are unexpectedly treated (low propensity but treated) receive high weight, and vice versa.

4.2.2 Theory

The Horvitz-Thompson IPW estimator (Horvitz and Thompson, 1952) for the ATE is:

$$\widehat{\text{ATE}}_{\text{IPW}} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{T_i Y_i}{\hat{e}(X_i)} - \frac{(1-T_i)Y_i}{1-\hat{e}(X_i)}\right) \tag{4.5}$$

The **stabilized** (Hájek) variant normalizes by the sum of weights, which reduces variance:

$$\widehat{\text{ATE}}_{\text{Hájek}} = \frac{\sum_i T_i Y_i/\hat{e}(X_i)}{\sum_i T_i/\hat{e}(X_i)} - \frac{\sum_i (1-T_i)Y_i/(1-\hat{e}(X_i))}{\sum_i (1-T_i)/(1-\hat{e}(X_i))} \tag{4.6}$$

Weight trimming is essential in practice: propensity scores near 0 or 1 produce extreme weights that inflate variance. Our implementation trims weights at the 1st and 99th percentiles.

4.2.3 Implementation

See `src/methods/classical/ipw.py`. The implementation uses stabilized (Hájek) weighting with configurable trimming thresholds. Bootstrap resampling provides inference. In our PKH experiment, IPW achieved an ATE of 5.60 with 1.05% relative bias and proper coverage, making it one of the best-performing classical methods.

4.3 Doubly Robust Estimation (AIPW)

4.3.1 Intuition

What if your propensity score model is slightly wrong? What if your outcome model is slightly wrong? The **Augmented Inverse Propensity Weighted** (AIPW) estimator, also called the **Doubly Robust** estimator (Robins et al., 1994), gives you a second chance: it produces consistent estimates if *either* the propensity score model *or* the outcome model is correctly specified.

4.3.2 Theory

The AIPW estimator combines IPW with outcome regression:

$$\widehat{\text{ATE}}_{\text{AIPW}} = \frac{1}{n}\sum_{i=1}^{n}\left[\hat{\mu}_1(X_i) - \hat{\mu}_0(X_i) + \frac{T_i(Y_i - \hat{\mu}_1(X_i))}{\hat{e}(X_i)} - \frac{(1-T_i)(Y_i - \hat{\mu}_0(X_i))}{1-\hat{e}(X_i)}\right] \tag{4.7}$$

where $\hat{\mu}_t(x) = \hat{\mathbb{E}}[Y \mid X = x, T = t]$ are outcome models for each treatment group, and $\hat{e}(x)$ is the estimated propensity score.

The key insight: if the outcome models $\hat{\mu}_t$ are correct, the IPW terms merely add noise; if the propensity model $\hat{e}$ is correct, the outcome model terms merely add efficiency. In either case, the estimator is consistent.

Additionally, AIPW can estimate individual-level effects:

$$\hat{\tau}_i^{\text{AIPW}} = \hat{\mu}_1(X_i) - \hat{\mu}_0(X_i) + \frac{T_i(Y_i - \hat{\mu}_1(X_i))}{\hat{e}(X_i)} - \frac{(1-T_i)(Y_i - \hat{\mu}_0(X_i))}{1-\hat{e}(X_i)} \tag{4.8}$$

4.3.3 Implementation

See `src/methods/classical/doubly_robust.py`. Both outcome and propensity models use `GradientBoostingRegressor` and `GradientBoostingClassifier` respectively. The file computes both ATE and individual CATE estimates. Bootstrap provides inference.

Note: Why Doubly Robust is Often Recommended

In our Monte Carlo simulation study (50 runs), AIPW achieved 86% coverage rate and 0.83% mean relative bias for PKH, among the best of all methods. The "double robustness" property provides valuable insurance against model misspecification. This is why many modern causal inference textbooks recommend AIPW as a default choice (Robins et al., 1994).

Chapter 5

Machine Learning Methods for Treatment Effect Estimation

Modern machine learning extends classical methods by offering flexible, nonparametric approaches to both average and heterogeneous treatment effect estimation. This chapter covers four families of ML-based causal methods.

5.1 Meta-Learners

Meta-learners (Künzel et al., 2019) are "wrappers" that convert any supervised learning algorithm into a causal inference method. The term "meta" reflects that the causal strategy is agnostic to the base learner.

5.1.1 S-Learner (Single Model)

The simplest approach: fit one model that includes treatment as a feature.

Step 1: Fit $\hat{\mu}(x,t) = \hat{\mathbb{E}}[Y \mid X = x, T = t]$ on the full dataset.

Step 2: Estimate CATE as:

$$\hat{\tau}_S(x) = \hat{\mu}(x,1) - \hat{\mu}(x,0) \tag{5.1}$$

Pros: Simple, uses all data. **Cons**: The model may "ignore" the treatment variable if it contributes little to prediction, a phenomenon called *regularization-induced confounding*.

5.1.2 T-Learner (Two Models)

Fit separate outcome models for each treatment group:

Step 1: Fit $\hat{\mu}_1(x)$ on $\{(X_i, Y_i) : T_i = 1\}$ and $\hat{\mu}_0(x)$ on $\{(X_i, Y_i) : T_i = 0\}$.

Step 2: $\hat{\tau}_T(x) = \hat{\mu}_1(x) - \hat{\mu}_0(x)$.

Pros: Allows fully different models per group. **Cons**: Does not share information between groups; can overfit with small treatment groups.

5.1.3 X-Learner (Cross-Learner)

The X-Learner (Künzel et al., 2019) improves on T-Learner by using cross-group imputation:

Step 1: Fit $\hat{\mu}_0$ and $\hat{\mu}_1$ as in T-Learner.

Step 2: Impute individual treatment effects:

$$\tilde{D}_i^1 = Y_i - \hat{\mu}_0(X_i) \quad \text{for } T_i = 1 \tag{5.2}$$

$$\tilde{D}_i^0 = \hat{\mu}_1(X_i) - Y_i \quad \text{for } T_i = 0 \tag{5.3}$$

Step 3: Fit $\hat{\tau}_1(x)$ on $\tilde{D}^1$ and $\hat{\tau}_0(x)$ on $\tilde{D}^0$.

Step 4: Combine using propensity score weighting:

$$\hat{\tau}_X(x) = \hat{e}(x) \cdot \hat{\tau}_0(x) + (1 - \hat{e}(x)) \cdot \hat{\tau}_1(x) \tag{5.4}$$

The X-Learner excels when treatment groups are imbalanced, as it leverages information from the larger group to improve estimates in the smaller group.

5.1.4 Implementation

All three meta-learners are implemented in `src/methods/ml_based/meta_learners.py` using `GradientBoostingRegressor` as the base learner. Each class (`SLearner`, `TLearner`, `XLearner`) follows a consistent interface with `estimate_ate()` returning both ATE and CATE estimates.

In our PKH experiments: S-Learner achieved the lowest ATE bias (0.52%), X-Learner achieved the best CATE correlation ($r = 0.67$), and T-Learner had the highest PEHE. In the Monte Carlo study, X-Learner achieved 90% coverage with the lowest $\sqrt{\text{PEHE}}$ among meta-learners (0.70), confirming its theoretical advantages.

5.2 Causal Forest

5.2.1 Theory

The Causal Forest (Wager and Athey, 2018; Athey et al., 2019) adapts random forests to directly estimate heterogeneous treatment effects. Unlike meta-learners that repurpose prediction algorithms, Causal Forest modifies the tree-splitting criterion to maximize treatment effect heterogeneity.

Each tree in the forest partitions the covariate space into leaves. Within each leaf L, the local treatment effect is estimated as:

$$\hat{\tau}(x) = \frac{\sum_{i \in L(x)} T_i Y_i / \hat{e}(X_i)}{\sum_{i \in L(x)} T_i / \hat{e}(X_i)} - \frac{\sum_{i \in L(x)} (1 - T_i) Y_i / (1 - \hat{e}(X_i))}{\sum_{i \in L(x)} (1 - T_i) / (1 - \hat{e}(X_i))} \tag{5.5}$$

The forest averages predictions across trees using **honesty**, separate subsamples for building the tree structure and estimating within-leaf effects which enables valid asymptotic confidence intervals.

5.2.2 The R-Learner Foundation

Our implementation uses EconML's `CausalForestDML`, which combines Causal Forest with the R-Learner framework. The R-Learner first "residualizes" both the outcome and treatment:

$$\tilde{Y}_i = Y_i - \hat{m}(X_i), \quad \hat{m}(x) = \hat{\mathbb{E}}[Y \mid X = x] \tag{5.6}$$

$$\tilde{T}_i = T_i - \hat{e}(X_i) \tag{5.7}$$

Then estimates $\tau(x)$ by regressing $\tilde{Y}$ on $\tilde{T}$ using the forest. This orthogonalization removes confounding bias from nuisance parameter estimation.

5.2.3 Implementation

See `src/methods/ml_based/causal_forest.py`. The implementation uses EconML's `CausalForestDML` with `GradientBoostingRegressor` for outcome and `GradientBoostingClassifier` for treatment nuisance models. The `discrete_treatment=True` flag is essential, it tells EconML that the treatment is binary, allowing proper use of a classifier for propensity estimation.

Note: Implementation Lesson

During development, we initially omitted the `discrete_treatment=True` parameter, causing EconML to raise an error: *"Cannot use a classifier as a first stage model when the target is continuous."* This taught us that EconML defaults to assuming continuous treatment, and the discrete flag must be explicitly set. Additionally, the inference objects returned by EconML change shape depending on the `discrete_treatment` setting, scalars vs. arrays, which required wrapping results with `np.atleast_1d()` for robust handling. These fixes are documented in the code comments.

A fallback implementation using manual T-Learner with Random Forest is provided when EconML is not installed.

5.3 Double Machine Learning (DML)

5.3.1 Theory

Double Machine Learning (Chernozhukov et al., 2018) provides a principled framework for using flexible ML methods in causal inference while maintaining valid statistical inference. The key innovation is **cross-fitting**: using out-of-fold predictions for nuisance parameters to avoid overfitting bias.

The DML procedure:

Step 1: Split data into K folds.

Step 2: For each fold k, use the remaining $K-1$ folds to estimate nuisance functions:

$$\hat{m}_{-k}(x) = \hat{\mathbb{E}}[Y \mid X = x] \quad \text{(outcome model)} \tag{5.8}$$

$$\hat{e}_{-k}(x) = \hat{\mathbb{E}}[T \mid X = x] \quad \text{(treatment model)} \tag{5.9}$$

Step 3: Compute residuals on fold k:

$$\tilde{Y}_i = Y_i - \hat{m}_{-k}(X_i) \tag{5.10}$$

$$\tilde{T}_i = T_i - \hat{e}_{-k}(X_i) \tag{5.11}$$

Step 4: Estimate the treatment effect by regressing $\tilde{Y}$ on $\tilde{T}$:

$$\hat{\theta}_{\text{DML}} = \frac{\sum_i \tilde{T}_i \tilde{Y}_i}{\sum_i \tilde{T}_i^2} \tag{5.12}$$

The cross-fitting ensures that the nuisance estimates are independent of the data used for effect estimation, yielding $\sqrt{n}$-consistent and asymptotically normal estimates even when nuisance functions are estimated with slow-converging ML methods.

5.3.2 Implementation

See `src/methods/ml_based/double_ml.py`. Uses EconML's `LinearDML` with 3-fold cross-fitting and gradient boosting nuisance models. Like Causal Forest, the `discrete_treatment=True` flag is required.

In our PKH experiment, DML achieved the best PEHE (0.61) and highest Pearson correlation with true effects ($r = 0.73$), suggesting strong heterogeneous effect recovery. However, its ATE coverage was only 22% in the Monte Carlo study, indicating that the confidence intervals may be too narrow, a known issue when the linear treatment effect assumption is violated.

Note: When Does DML Shine?

DML is theoretically optimal when the treatment effect is approximately linear in covariates (the "partially linear" model). For highly nonlinear treatment effect surfaces, Causal Forest or X-Learner may be preferable. Our results confirm this: DML achieves the best PEHE but poor coverage, suggesting it captures heterogeneity well but underestimates uncertainty.

Chapter 6

Deep Learning Methods for Causal Inference

Deep learning brings powerful representation learning to causal inference, potentially capturing complex nonlinear relationships that simpler methods miss. However, as our experiments reveal, this power comes with significant practical challenges.

6.1 Causal Effect Variational Autoencoder (CEVAE)

6.1.1 Intuition

CEVAE (Louizos et al., 2017) addresses a fundamental challenge: what if treatment assignment depends on variables we cannot observe? CEVAE posits a latent variable z that simultaneously influences covariates x, treatment t, and outcome y. By learning this latent structure, CEVAE attempts to "reconstruct" unobserved confounders.

6.1.2 Architecture

The model has two components:

Encoder $q_\phi(z \mid x, t, y)$: Infers the posterior distribution of the latent confounder given all observed variables. The encoder outputs parameters μ_z and σ_z^2 of a Gaussian posterior:

$$q_\phi(z \mid x, t, y) = \mathcal{N}(z; \mu_\phi(x, t, y), \sigma_\phi^2(x, t, y)) \tag{6.1}$$

Decoder $p_\theta(x, t, y \mid z)$: Reconstructs observed variables from the latent representation. Critically, the decoder has *separate outcome heads* for $t = 0$ and $t = 1$:

$$p_\theta(y \mid z, t) = \begin{cases} f_{\theta_0}(z) & \text{if } t = 0 \\ f_{\theta_1}(z) & \text{if } t = 1 \end{cases} \tag{6.2}$$

6.1.3 Training Objective

CEVAE is trained by maximizing the Evidence Lower Bound (ELBO):

$$\mathcal{L} = \mathbb{E}_{q_\phi(z|x,t,y)} \left[\log p_\theta(x, t, y \mid z)\right] - \mathrm{KL}\left[q_\phi(z|x, t, y) \| p(z)\right] \tag{6.3}$$

The first term encourages accurate reconstruction; the second term regularizes the latent space toward a standard normal prior $p(z) = \mathcal{N}(0, I)$.

6.1.4 Estimating Treatment Effects

After training, CATE is estimated via Monte Carlo sampling:

$$\hat{\tau}(x_i) = \frac{1}{M} \sum_{m=1}^{M} \left[f_{\theta_1}(z_i^{(m)}) - f_{\theta_0}(z_i^{(m)}) \right], \quad z_i^{(m)} \sim q_\phi(z \mid x_i, t_i, y_i) \tag{6.4}$$

6.1.5 Implementation

See `src/methods/deep_learning/cevae.py`. The implementation uses PyTorch with a two-layer encoder and decoder (hidden dimension 64, latent dimension 20). Training runs for 100 epochs with batch size 128, optimized with Adam ($lr = 10^{-3}$).

Note: CEVAE Performance and Lessons Learned

CEVAE showed the weakest performance among all methods, with 73% relative bias on PKH and 90% on JKN. This is a valuable finding, not a failure. Several factors contribute: (1) The small network architecture (necessary for CPU training) limits expressiveness; (2) 100 epochs may be insufficient for convergence on complex data; (3) The VAE's ELBO objective does not directly optimize for treatment effect accuracy. During development, we also discovered that CEVAE's ATE and confidence intervals were computed inconsistently , the point estimate used the posterior mean while the CI used Monte Carlo samples from a different distribution. We corrected this to use Monte Carlo consistently for both, ensuring the CI always contains the point estimate. This debugging journey reinforces a key lesson: **deep learning for causal inference requires careful validation**, not blind trust in neural network outputs.

6.2 Counterfactual Regression Network (CFRNet)

6.2.1 Intuition

CFRNet (Shalit et al., 2017) takes a representation learning approach: learn a feature representation $\Phi(x)$ in which the treated and control distributions are similar, then estimate outcomes in this balanced representation space.

6.2.2 Architecture

CFRNet has three components:

Shared Representation $\Phi : \mathcal{X} \to \mathcal{R}$: Maps raw covariates to a learned representation.

Outcome Heads $h_0, h_1 : \mathcal{R} \to \mathbb{R}$: Separate networks predicting outcomes for control and treated groups.

The CATE estimate is:

$$\hat{\tau}(x) = h_1(\Phi(x)) - h_0(\Phi(x)) \tag{6.5}$$

6.2.3 The IPM Regularization

The key innovation is a regularization term that penalizes distributional imbalance between treated and control representations. CFRNet uses the **Maximum Mean Discrepancy (MMD)**:

$$\text{MMD}^2(\hat{p}_1, \hat{p}_0) = \left\| \frac{1}{n_1} \sum_{i:T_i=1} \phi(\Phi(X_i)) - \frac{1}{n_0} \sum_{i:T_i=0} \phi(\Phi(X_i)) \right\|_{\mathcal{H}}^2 \tag{6.6}$$

where ϕ is a kernel feature map. Intuitively, this pushes the network to learn representations where treated and control groups "look the same," eliminating selection bias in the representation space.

The total loss is:

$$\mathcal{L} = \underbrace{\frac{1}{n} \sum_i (Y_i - h_{T_i}(\Phi(X_i)))^2}_{\text{prediction loss}} + \alpha \cdot \underbrace{\text{MMD}^2(\hat{p}_1, \hat{p}_0)}_{\text{balance regularization}} \tag{6.7}$$

6.2.4 Implementation

See `src/methods/deep_learning/cfrnet.py`. Architecture: 3-layer shared representation (input $\to$ 64 $\to$ 64), 2-layer outcome heads per treatment group, dropout ($p = 0.1$) for uncertainty, MMD weight $\alpha = 1.0$. Training: 150 epochs, batch size 128, Adam optimizer.

CFRNet performed significantly better than CEVAE: 2.61% relative bias on PKH and 1.99% on JKN. However, its uncertainty quantification was limited (SE ≈ 0) because dropout-based uncertainty can collapse when the network is confident. For production use, we recommend combining CFRNet's point estimates with bootstrap-based confidence intervals.

Chapter 7

Policy Evaluation Methods for Panel Data

The provincial minimum wage dataset requires fundamentally different methods than the cross-sectional PKH and JKN datasets. Panel data, repeated observations of the same units over time, enables methods that exploit temporal variation to identify causal effects.

7.1 Difference-in-Differences (DiD)

7.1.1 Intuition

DiD compares the *change* in outcomes between treated and control groups, before and after treatment. By differencing out unit-specific and time-specific effects, DiD eliminates confounders that are constant over time or constant across units.

7.1.2 Two-Way Fixed Effects (TWFE) Model

The standard TWFE specification is:

$$Y_{it} = \alpha_i + \lambda_t + \delta \cdot D_{it} + \varepsilon_{it} \tag{7.1}$$

where α_i are province fixed effects (absorbing time-invariant province characteristics), λ_t are year fixed effects (absorbing common shocks), D_{it} is the treatment indicator, and δ is the DiD estimate of the ATT.

Standard errors are clustered at the province level to account for serial correlation within provinces:

$$\hat{V}_{\text{cluster}} = (X'X)^{-1} \left(\sum_{g=1}^{G} X_g' \hat{u}_g \hat{u}_g' X_g \right) (X'X)^{-1} \tag{7.2}$$

7.1.3 The Parallel Trends Assumption

DiD's identification relies on the **parallel trends assumption**: in the absence of treatment, treated and control units would have followed parallel outcome trajectories:

$$\mathbb{E}[Y_{it}(0) - Y_{i,t-1}(0) \mid D_i = 1] = \mathbb{E}[Y_{it}(0) - Y_{i,t-1}(0) \mid D_i = 0] \tag{7.3}$$

While untestable for the post-treatment period, we can assess its plausibility by testing for *differential pre-trends* using an event study specification:

$$Y_{it} = \alpha_i + \lambda_t + \sum_{k \neq -1} \gamma_k \cdot \mathbf{1}[t - t_i^* = k] + \varepsilon_{it} \tag{7.4}$$

where t_i^* is unit i's treatment date and k indexes event time. Pre-treatment coefficients γ_k for $k < 0$ should be jointly insignificant if parallel trends holds.

7.1.4 Staggered Adoption

Our provincial dataset features **staggered treatment adoption**: different provinces implement aggressive wage increases at different times. Recent econometric literature (Callaway and Sant'Anna, 2021) has shown that TWFE can produce biased estimates under staggered adoption due to "forbidden comparisons", using already-treated units as controls. Our event study specification mitigates this by explicitly modeling dynamic effects.

7.1.5 Implementation

See `src/methods/policy_evaluation/did.py`. The implementation provides: (1) TWFE estimation with cluster-robust standard errors via `statsmodels`; (2) Event study with configurable pre/post windows; (3) Pre-trend test via joint F-test of pre-treatment coefficients.

Our results: TWFE ATT $= -2.95$ percentage points ($p < 0.001$), suggesting that aggressive minimum wage increases reduced employment by about 3 percentage points. The event study pre-trend test p-value of 0.65 supports the parallel trends assumption.

Note: Interpreting the Provincial Results

The event study plot (`results/provincial/provincial_event_study.png`) shows flat pre-treatment coefficients and a clear negative break at treatment, providing compelling visual evidence for the causal interpretation. However, with only 6 treated provinces, precision is inherently limited.

7.2 Synthetic Control Method (SCM)

7.2.1 Intuition

When the number of treated units is very small (sometimes just one), traditional regression methods lack statistical power. The Synthetic Control Method (Abadie et al., 2010) constructs a "synthetic" version of the treated unit from a weighted combination of control ("donor") units that mimics the treated unit's pre-treatment trajectory.

7.2.2 Theory

For treated unit $j = 1$ with treatment at time T_0, we seek weights $w_2, \ldots, w_J$ such that:

$$\min_{w} \sum_{t=1}^{T_0} \left(Y_{1t} - \sum_{j=2}^{J} w_j Y_{jt} \right)^2 \tag{7.5}$$

subject to $w_j \geq 0$ and $\sum_j w_j = 1$.

The treatment effect is estimated as the gap between the actual treated unit and its synthetic counterpart:

$$\hat{\tau}_t = Y_{1t} - \sum_{j=2}^{J} \hat{w}_j Y_{jt}, \quad t > T_0 \tag{7.6}$$

7.2.3 Placebo Inference

Since we have only one treated unit, classical inference is impossible. Instead, SCM uses **placebo tests**: apply the same procedure to every control unit (pretending each was treated). If the treated unit's effect is unusually large relative to placebo effects, we have evidence of a genuine treatment effect.

The placebo p-value is:

$$p = \frac{\text{rank of treated unit's RMSPE ratio among all units}}{J} \tag{7.7}$$

where RMSPE is the root mean squared prediction error (post-treatment / pre-treatment).

7.2.4 Implementation

See `src/methods/policy_evaluation/synthetic_control.py`. The implementation optimizes donor weights using `scipy.optimize.minimize` with SLSQP, computes pre-treatment RMSPE as a fit diagnostic, and runs full placebo inference across all donor provinces.

Our results for Banten province: ATT $= -1.47$ percentage points, pre-treatment RMSPE $= 0.22$ (good fit), placebo p-value $= 0.26$. The top donor provinces contributing to synthetic Banten are Jawa Barat (39.4%), Riau (31.7%), and Bengkulu (16.3%).

Note: Synthetic Control Visualization

The synthetic control plot (`results/provincial/provincial_synthetic_control.png`) shows the actual Banten trajectory alongside its synthetic counterpart. A clear divergence after 2018 (treatment year) provides visual evidence of the policy impact. The gap plot below shows the treatment effect trajectory over time.

Chapter 8

Project Architecture: A Complete Code Guide

This chapter maps every file in the project to its role, so that readers can navigate the codebase with full understanding of the workflow.

8.1 Directory Structure

The project follows a modular architecture separating data generation, methods, evaluation, and execution:

```
indonesian_causal_inference/
+-- configs/
|   +-- config.yaml              [A]
+-- data/                         [B]
|   +-- pkh_observed.csv
|   +-- pkh_ground_truth.csv
|   +-- jkn_observed.csv
|   +-- jkn_ground_truth.csv
|   +-- provincial_panel.csv
|   +-- provincial_treatment_info.csv
+-- src/
|   +-- data_generation/          [C]
|   +-- preprocessing/            [D]
|   +-- methods/                  [E]
|   |   +-- classical/
|   |   +-- ml_based/
|   |   +-- deep_learning/
|   |   +-- policy_evaluation/
|   +-- evaluation/               [F]
|   +-- utils/                    [G]
+-- results/                      [H]
+-- generate_datasets.py          [Step 1]
+-- run_experiments.py            [Step 2]
+-- run_simulation_study.py       [Step 3]
+-- requirements.txt
```

```
+-- README.md
```

8.2 Component Descriptions

[A] **Configuration** (`configs/config.yaml`): Central configuration file controlling sample sizes, random seeds, number of bootstrap iterations, training epochs, and experiment parameters. Modify this file to change experiment settings without touching any Python code.

[B] **Data Directory** (`data/`): Contains all six CSV files after running Step 1. The "observed" files contain what a researcher would see in practice (covariates, treatment, outcome). The "ground truth" files contain individual-level true effects used only for evaluation.

[C] **Data Generation** (`src/data_generation/`): Three scripts that create the datasets:

- `generate_pkh_dataset.py`: Generates 3,000 household records with 25 covariates, PMT-based treatment assignment, and heterogeneous welfare effects.
- `generate_jkn_dataset.py`: Generates 10,000 maternal records in 5,000 pairs with 30 covariates and neonatal health outcomes.
- `generate_provincial_dataset.py`: Generates 34 provinces $\times$ 15 years panel data with staggered minimum wage treatment.

[D] **Preprocessing** (`src/preprocessing/data_loader.py`): A unified `DataLoader` class that loads any of the three datasets with optional normalization. Returns feature matrices, treatment vectors, outcomes, and ground truth in a consistent format. This abstraction ensures all methods receive data in an identical format.

[E] **Methods** (`src/methods/`): The core of the project. Each method file implements a class with a consistent `estimate_ate(X, T, Y, n_bootstrap)` interface returning a dictionary with keys: `method`, `ate_estimate`, `se`, `ci_lower`, `ci_upper`, and optionally `cate`. This uniform interface makes methods interchangeable and simplifies the experiment runner.

[F] **Evaluation** (`src/evaluation/`):

- `metrics.py`: Computes all evaluation metrics (ATE bias, relative bias, coverage, PEHE, correlations, policy overlap). The `evaluate_method()` function takes a method result and ground truth, returning a comprehensive metrics dictionary. The `compare_methods()` function ranks methods by performance.
- `visualization.py`: Generates plots including forest plots (`plot_ate_comparison`), CATE scatter plots (`plot_cate_distribution`), propensity overlap histograms (`plot_propensity_overlap`), covariate balance Love plots (`plot_covariate_balance`), event study plots (`plot_event_study`), and synthetic control gap plots (`plot_synthetic_control`

[G] Utilities (`src/utils/helpers.py`): Helper functions including `set_seed()` for reproducibility, `load_config()` for YAML parsing, a `@timer` decorator for profiling, and `print_metrics_table()` for formatted console output.

[H] Results (`results/`): Output directory populated by Steps 2 and 3 with subdirectories for each experiment containing CSV result tables and PNG visualization files.

8.3 Method Interface Design

Every causal method follows the same pattern, enabling easy extension:

1. Accept inputs: `X` (covariates, $n \times p$ array), `T` (treatment, binary vector), `Y` (outcome, continuous vector), `n_bootstrap` (number of bootstrap iterations for inference).
2. Return a dictionary: `{'method': str, 'ate_estimate': float, 'se': float, 'ci_lower': float, 'ci_upper': float, 'cate': array (optional)}`.

To add a new method, create a class following this interface in the appropriate subdirectory and add the corresponding call in `run_experiments.py`.

Chapter 9

Running the Experiments: Step-by-Step Guide

This chapter provides exact instructions for reproducing all results.

9.1 Prerequisites

9.1.1 System Requirements

Resource	Minimum	Recommended
RAM	4 GB	8 GB
Disk Space	500 MB	1 GB
CPU	2 cores	4+ cores
GPU	Not needed	Not needed
Python	3.9+	3.10+

Table 9.1: System requirements. All experiments run on a standard laptop without GPU.

9.1.2 Installation

Create a virtual environment and install dependencies:

```
python -m venv venv
source venv/bin/activate        # Linux/Mac
venv\Scripts\activate           # Windows

pip install -r requirements.txt
pip install torch --index-url https://download.pytorch.org/whl/cpu
```

The CPU-only PyTorch installation saves significant disk space (∼200 MB vs. ∼2 GB for the full CUDA version) with no performance impact for our small-scale models.

9.2 Step 1: Generate Datasets (∼5 seconds)

```
python generate_datasets.py
```

This creates all six CSV files in `data/`. Generation is deterministic (seed = 42): every run produces identical files. The script prints summary statistics for each dataset (sample sizes, treatment rates, feature counts).

9.3 Step 2: Run Experiments (∼95 minutes total)

```
python run_experiments.py                # All experiments
python run_experiments.py --pkh          # PKH only (~20 min)
python run_experiments.py --jkn          # JKN only (~60 min)
python run_experiments.py --provincial   # Provincial only (~2 min)
```

For each dataset, the script:

1. Loads data via the `DataLoader` (with normalization for PKH/JKN).
2. Runs all applicable methods sequentially, printing ATE, SE, and wall-clock time for each.
3. Computes comprehensive evaluation metrics comparing estimated effects to ground truth.
4. Generates and saves all visualization plots.
5. Saves result tables as CSV files.

Expected output files:

File	Description
`results/pkh/pkh_results.csv`	ATE and ITE metrics for all PKH methods
`results/pkh/pkh_ate_comparison.png`	Forest plot comparing ATE estimates
`results/pkh/pkh_cate_scatter.png`	Estimated vs. true ITE scatter plots
`results/jkn/jkn_results.csv`	ATE and ITE metrics for all JKN methods
`results/jkn/jkn_ate_comparison.png`	Forest plot for JKN
`results/jkn/jkn_cate_scatter.png`	CATE scatter plots for JKN
`results/provincial/provincial_did.png`	DiD event study plot
`results/provincial/provincial_event_study.png`	Dynamic treatment effect coefficients
`results/provincial/provincial_synthetic_control.png`	Actual vs. synthetic trajectories

Table 9.2: Output files generated by `run_experiments.py`. Insert these figures where indicated in this document.

9.4 Step 3: Monte Carlo Simulation (∼4–5 hours per dataset)

```
python run_simulation_study.py --n-sims 10          # Quick test
python run_simulation_study.py --n-sims 50          # Full study (PKH)
python run_simulation_study.py --dataset jkn --n-sims 50  # JKN study
```

The simulation study regenerates datasets with different random seeds and runs all fast methods (excluding deep learning for computational tractability) on each realization. This produces the metrics reported in Chapter 10: mean bias, coverage rate, and $\sqrt{\text{PEHE}}$ averaged across seeds.

Note: Runtime Expectations

On our test system (Intel Core i7, 16 GB RAM), the 50-simulation PKH study completed in approximately 4 hours 23 minutes. The JKN study took 4 hours 8 minutes. The majority of time is spent on bootstrap confidence intervals for classical and meta-learner methods. Reducing `n_bootstrap` from 200 (full experiment) to 50 (simulation study) significantly improves throughput.

Chapter 10

Results and Analysis

This chapter presents the complete results from our experiments and simulation study, organized by dataset.

10.1 PKH Results: Cross-Sectional Benchmark

10.1.1 Single-Run ATE Comparison

Table 10.1 summarizes ATE estimates from all 10 methods applied to the PKH dataset (true ATE = 5.538).

Method	ATE	Bias	Rel. Bias	SE	95% CI	Covers
S-Learner	5.567	0.029	0.52%	0.054	[5.47, 5.68]	Yes
IPW	5.596	0.058	1.05%	0.059	[5.50, 5.73]	Yes
X-Learner	5.617	0.079	1.43%	0.050	[5.54, 5.73]	Yes
Double ML	5.412	0.126	2.28%	0.059	[5.30, 5.53]	No
AIPW	5.674	0.135	2.44%	0.054	[5.59, 5.79]	No
CFRNet	5.394	0.145	2.61%	—	—	No
T-Learner	5.690	0.152	2.74%	0.055	[5.60, 5.81]	No
Causal Forest	5.379	0.159	2.88%	0.310	[4.77, 5.99]	Yes
PSM	5.765	0.227	4.09%	0.123	[5.57, 6.04]	No
CEVAE	1.496	4.043	73.0%	0.003	[1.49, 1.50]	No

Table 10.1: PKH ATE estimates from a single experimental run. Methods ranked by absolute bias.

10.1.2 Heterogeneous Effect Recovery (ITE Metrics)

Table 10.2 shows how well each method recovers individual-level treatment effects.

Method	$\sqrt{\text{PEHE}}$	Norm. PEHE	Pearson r	Spearman ρ	Policy Overlap
Double ML	0.610	0.698	0.731	0.728	0.775
X-Learner	0.693	0.794	0.669	0.669	0.749
Causal Forest	0.709	0.811	0.683	0.679	0.756
S-Learner	0.892	1.022	0.543	0.539	0.687
CFRNet	1.033	1.183	0.551	0.554	0.701
T-Learner	1.276	1.461	0.433	0.417	0.649
CEVAE	4.126	4.725	0.384	0.374	0.611

Table 10.2: PKH ITE-level metrics. Lower $\sqrt{\text{PEHE}}$ and higher correlations indicate better CATE recovery.

Key Finding: Double ML achieves the best CATE recovery ($\sqrt{\text{PEHE}} = 0.61$, $r = 0.73$), followed by X-Learner and Causal Forest. This suggests that the residualization strategy of DML effectively isolates treatment effect heterogeneity in the PKH data.

10.1.3 Monte Carlo Simulation Results (50 Seeds)

Table 10.3 presents aggregated results across 50 independent replications, providing reliable performance estimates.

Method	Mean ATE	SD	Mean Bias	Coverage	Mean $\sqrt{\text{PEHE}}$
IPW	5.517	0.050	0.039	**94%**	—
X-Learner	5.527	0.051	0.039	90%	**0.702**
AIPW	5.542	0.053	0.046	86%	—
T-Learner	5.551	0.061	0.056	78%	1.267
S-Learner	5.445	0.057	0.075	58%	0.875
Double ML	5.366	0.052	0.152	22%	0.604
PSM	5.743	0.072	0.225	38%	—

Table 10.3: PKH Monte Carlo results (50 seeds). Coverage is for 95% CI. Bold indicates best per column.

Key Findings:

1. **IPW** achieves the best coverage (94%), closest to the nominal 95%. This validates the propensity-based approach for the PKH targeting mechanism.

2. **X-Learner** offers the best combination of low bias and strong CATE recovery ($\sqrt{\text{PEHE}} = 0.70$) with good coverage (90%).

3. **Double ML** has the best individual $\sqrt{\text{PEHE}}$ (0.60) but severely under-covers (22%), suggesting its confidence intervals are too narrow for this data.

4. **AIPW**'s double robustness provides reliable performance (0.83% mean relative bias, 86% coverage).

10.2 JKN Results

The JKN dataset (10,000 samples, true ATE = 8.122) provides a larger-scale test. Table 10.4 summarizes single-run ATE estimates.

Method	ATE	Bias	Rel. Bias	SE	95% CI	Covers
AIPW	8.050	0.073	0.89%	0.065	[7.91, 8.15]	Yes
IPW	8.049	0.073	0.90%	0.070	[7.91, 8.18]	Yes
X-Learner	8.047	0.076	0.93%	0.059	[7.93, 8.16]	Yes
T-Learner	8.030	0.092	1.13%	0.054	[7.93, 8.15]	Yes
PSM	8.000	0.122	1.51%	0.079	[7.90, 8.21]	Yes
S-Learner	7.981	0.142	1.74%	0.056	[7.84, 8.07]	No
CFRNet	8.284	0.162	1.99%	—	—	No
Double ML	7.940	0.182	2.25%	0.064	[7.81, 8.07]	No
Causal Forest	7.933	0.189	2.33%	0.562	[6.83, 9.04]	Yes
CEVAE	0.805	7.317	90.1%	0.004	[0.80, 0.81]	No

Table 10.4: JKN ATE estimates. Methods ranked by absolute bias.

The larger sample size generally improves all methods. AIPW and IPW again lead in ATE accuracy, while most classical and ML methods achieve coverage. CEVAE continues to struggle severely, reinforcing that its architecture needs substantially more capacity and training to handle this data scale.

10.3 Provincial Results: Panel Data Methods

10.3.1 Difference-in-Differences

The TWFE DiD estimate:

- **ATT**: -2.945 percentage points
- **Cluster-robust SE**: 0.600
- **95% CI**: $[-4.12, -1.77]$
- p**-value**: < 0.001

The event study confirms parallel pre-trends ($p = 0.65$ for joint pre-trend test), with pre-treatment coefficients statistically indistinguishable from zero.

10.3.2 Synthetic Control

For Banten province (first treated unit, treatment year 2018):

- **ATT**: -1.474 percentage points

- **Pre-treatment RMSPE**: 0.224 (excellent pre-treatment fit)
- **Top donors**: Jawa Barat (39.4%), Riau (31.7%), Bengkulu (16.3%)
- **Placebo p-value**: 0.265 (rank 9 of 34)

The placebo p-value of 0.265 does not reach conventional significance, suggesting that while there is a negative effect, it is not extreme relative to placebo variation across provinces. This illustrates a known limitation of SCM: with few treated units, statistical power is limited.

10.4 Cross-Dataset Insights

Comparing results across all three datasets reveals important patterns:

1. **No single method dominates everywhere**. IPW excels at coverage, X-Learner at CATE recovery, AIPW at balanced performance. Method selection should depend on the research question.
2. **Simple methods are competitive**. IPW (a 1950s invention) outperforms deep learning on both datasets. Complexity does not guarantee accuracy.
3. **Deep learning needs scale**. CEVAE's failure is instructive: with limited data and compute, deep learning methods can dramatically underperform classical approaches. CFRNet performs much better, likely due to its simpler architecture and direct prediction loss.
4. **Coverage matters**. Double ML has the best PEHE but 22% coverage, meaning it gives precise but overconfident estimates. For policy applications, reliable uncertainty quantification (as provided by IPW and AIPW) may be more valuable than point estimate accuracy.
5. **Panel methods require different diagnostics**. The provincial analysis relies on parallel trends tests and placebo inference rather than PEHE, reflecting the fundamentally different identification strategy.

Chapter 11

The Debugging Journey: Lessons from Building This Project

Real research involves mistakes, corrections, and iterative improvement. This chapter documents the key technical challenges we encountered and how we resolved them, serving both as a cautionary guide and as evidence of rigorous development methodology.

11.1 The `discrete_treatment` Discovery

Problem: When running the Causal Forest method, EconML raised:

```
AttributeError: Cannot use a classifier as a first
stage model when the target is continuous!
```

Root Cause: EconML's `CausalForestDML` and `LinearDML` default to assuming continuous treatment. When we passed a `GradientBoostingClassifier` as the treatment model (appropriate for binary treatment), EconML tried to use its `.predict()` method in a regression context, which fails.

Fix: Adding `discrete_treatment=True` to both `CausalForestDML` and `LinearDML` constructors. This flag tells EconML to use the classifier's `.predict_proba()` method instead.

Lesson: Library defaults may not match your use case. Always read the API documentation carefully, especially for parameters that change fundamental model behavior.

11.2 The Scalar vs. Array Inference Bug

Problem: After fixing the `discrete_treatment` issue, a new error appeared:

```
IndexError: invalid index to scalar variable.
```

Root Cause: With `discrete_treatment=True`, EconML's inference objects return scalar values (since there is only one treatment contrast), but our code assumed array-shaped returns with indexing like `stderr_mean[0]`.

Fix: Wrapping all inference results with `np.atleast_1d()` before indexing, making the code robust to both scalar and array returns.

Lesson: When a library parameter changes model structure, downstream API returns may also change shape. Defensive programming (handling multiple return types) prevents cascading failures.

11.3 The CEVAE Confidence Interval Inconsistency

Problem: CEVAE reported ATE = 1.42 with CI = [1.49, 1.50]. The confidence interval did not contain the point estimate, a logical impossibility.

Root Cause: The ATE was computed using the posterior mean $z = \mu_\phi$, while the CI was computed from Monte Carlo samples $z \sim \mathcal{N}(\mu_\phi, \sigma_\phi^2)$. These two computation paths can produce systematically different estimates because the MC samples explore the full posterior while the mean collapses it.

Fix: We unified both computations to use Monte Carlo sampling: the final ATE is the mean of MC-sampled ATEs, and the CATE is the element-wise mean of MC-sampled CATEs. This ensures perfect consistency between point estimates and uncertainty bands.

Lesson: When using probabilistic models, ensure that point estimates and uncertainty measures are computed from the same procedure. Mixing deterministic and stochastic computation paths is a common source of inconsistency.

11.4 The Negative Error Bar Crash

Problem: Matplotlib's `errorbar` raised:

```
ValueError: 'xerr' must not contain negative values
```

Root Cause: The CEVAE CI inconsistency caused `xerr_lower = ATE - CI_lower` to be negative. While we fixed the root cause in CEVAE, we also added defensive clamping in the visualization code.

Fix: `xerr_lower = [max(0, a - l) for a, l in zip(ates, ci_lowers)]`.

Lesson: Defense in depth. Even after fixing the root cause, downstream code should handle edge cases gracefully. A visualization function should never crash due to unexpected input from an upstream method.

Note: Why We Report These Bugs

Reporting debugging journeys serves multiple purposes: (1) it demonstrates honest scientific practice; (2) it helps future researchers avoid the same pitfalls; (3) it shows technical problem-solving skills to recruiters; (4) it reinforces that good research is iterative, not linear. Every published paper has a hidden history of bugs, dead ends, and corrections, we choose to make ours visible.

Chapter 12

Limitations and Future Directions

12.1 Current Limitations

12.1.1 Data Limitations

1. **Sample Size Constraints**: The PKH dataset (3,000 households) and provincial panel (510 observations) are relatively small by modern standards. Larger datasets would enable better evaluation of data-hungry methods like CEVAE.

2. **Covariate Coverage**: While our datasets include 25–30 covariates each, real-world targeting data may include hundreds of variables. The unconfoundedness assumption becomes more plausible with richer covariates.

3. **Static Treatment**: All three experiments assume a single treatment assignment. In reality, PKH enrollment may change over time, JKN coverage expands, and minimum wage adjustments are continuous.

12.1.2 Methodological Limitations

1. **Unconfoundedness is Untestable**: All cross-sectional methods (Chapters 4–6) assume that all confounders are observed. If unmeasured confounders exist, all estimates may be biased. Sensitivity analyses (Rosenbaum, 2002) could quantify robustness to hidden bias.

2. **SUTVA Violations**: Social protection programs may have spillover effects (e.g., PKH recipients sharing resources with neighbors, minimum wage effects crossing provincial borders). Our analysis assumes no interference between units.

3. **Deep Learning Underperformance**: CEVAE's poor results may reflect insufficient architecture capacity rather than a fundamental limitation. GPU-based training with larger networks, more epochs, and hyperparameter tuning could significantly improve performance.

4. **Limited Staggered DiD Methodology**: Our TWFE estimator may be biased under heterogeneous treatment effects with staggered adoption (Callaway and Sant'Anna, 2021). More robust estimators (e.g., Callaway–Sant'Anna, Sun–Abraham) could be implemented.

5. **Bootstrap Inference**: Several methods rely on bootstrap confidence intervals, which can be computationally expensive and may not achieve exact coverage in finite samples.

12.1.3 Scope Limitations

1. **Binary Treatment Only**: All cross-sectional analyses assume binary treatment. Extensions to continuous treatment (dose-response) or multiple treatments are not covered.

2. **Single Outcome**: Each experiment focuses on one primary outcome. Multi-outcome analysis could reveal trade-offs (e.g., minimum wage effects on employment vs. poverty simultaneously).

3. **No Causal Discovery**: This project assumes the causal structure is known. Causal discovery methods (e.g., PC algorithm, FCI) could learn causal graphs from data.

12.2 Future Research Directions

12.2.1 Methodological Extensions

1. **Sensitivity Analysis**: Implement Rosenbaum bounds or the E-value framework (VanderWeele and Ding, 2017) to assess how robust conclusions are to unmeasured confounding.

2. **Advanced Staggered DiD**: Implement Callaway and Sant'Anna's (2021) group-time ATT estimator and Sun and Abraham's (2021) interaction-weighted estimator for more robust panel analysis.

3. **Continuous Treatment**: Extend to generalized propensity score methods for dose-response analysis of minimum wage levels.

4. **Causal Discovery Integration**: Combine treatment effect estimation with causal graph learning to identify which variables are confounders versus mediators.

5. **Bayesian Approaches**: Implement BART (Bayesian Additive Regression Trees) (Hill, 2011) and Bayesian Causal Forest (Hahn et al., 2020) for principled uncertainty quantification.

12.2.2 Applied Extensions

1. **Optimal Policy Learning**: Use estimated CATEs to design optimal targeting policies, identifying which households should receive PKH for maximum welfare impact per rupiah.

2. **Cost-Effectiveness Analysis**: Combine treatment effect estimates with program costs to evaluate cost-effectiveness across programs.

3. **Temporal Dynamics**: Model how treatment effects evolve over time using longitudinal extensions of CATE estimation.

4. **Geographic Heterogeneity Mapping**: Use CATE estimates to create province-level maps of treatment effect variation, informing spatially targeted policy.

5. **Transfer to Other Countries**: Adapt the benchmark framework for social protection programs in other developing countries (e.g., Brazil's Bolsa Família, India's MGNREGA).

12.2.3 Technical Extensions

1. **GPU-Accelerated Deep Learning**: Train CEVAE and CFRNet on GPU with larger architectures (hidden dim 256+, 500+ epochs) to properly evaluate deep learning potential.

2. **Ensemble Methods**: Combine predictions from multiple causal methods using super-learner or stacking approaches.

3. **Conformal Inference**: Apply conformal prediction to obtain distribution-free prediction intervals for individual treatment effects.

4. **Interactive Dashboard**: Build a Streamlit or Dash application for interactive exploration of treatment effect heterogeneity.

Bibliography

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701.

Holland, P. W. (1986). Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960.

Rosenbaum, P. R. and Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55.

Rosenbaum, P. R. (2002). *Observational Studies*. Springer, 2nd edition.

LaLonde, R. J. (1986). Evaluating the econometric evaluations of training programs with experimental data. *American Economic Review*, 76(4):604–620.

Horvitz, D. G. and Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685.

Robins, J. M., Rotnitzky, A., and Zhao, L. P. (1994). Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association*, 89(427):846–866.

Künzel, S. R., Sekhon, J. S., Bickel, P. J., and Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10):4156–4165.

Wager, S. and Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242.

Athey, S., Tibshirani, J., and Wager, S. (2019). Generalized random forests. *Annals of Statistics*, 47(2):1148–1178.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68.

Nie, X. and Wager, S. (2021). Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2):299–319.

Curth, A. and van der Schaar, M. (2021). On inductive biases for heterogeneous treatment effect estimation. *Advances in Neural Information Processing Systems*, 34.

Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R., and Welling, M. (2017). Causal effect inference with deep latent-variable models. *Advances in Neural Information Processing Systems*, 30.

Shalit, U., Johansson, F. D., and Sontag, D. (2017). Estimating individual treatment effect: Generalization bounds and algorithms. *International Conference on Machine Learning (ICML)*, pages 3076–3085.

Hill, J. L. (2011). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240.

Hahn, P. R., Murray, J. S., and Carvalho, C. M. (2020). Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects. *Bayesian Analysis*, 15(3):965–1056.

Abadie, A., Diamond, A., and Hainmueller, J. (2010). Synthetic control methods for comparative case studies: Estimating the effect of California's tobacco control program. *Journal of the American Statistical Association*, 105(490):493–505.

Callaway, B. and Sant'Anna, P. H. C. (2021). Difference-in-differences with multiple time periods. *Journal of Econometrics*, 225(2):200–230.

Sun, L. and Abraham, S. (2021). Estimating dynamic treatment effects in event studies with heterogeneous treatment effects. *Journal of Econometrics*, 225(2):175–199.

Alatas, V., Banerjee, A., Hanna, R., Olken, B. A., and Tobias, J. (2012). Targeting the poor: Evidence from a field experiment in Indonesia. *American Economic Review*, 102(4):1206–1240.

Cahyadi, E. R., Hanna, R., Olken, B. A., Prima, R. A., Satriawan, E., and Syamsulhakim, E. (2020). Cumulative impacts of conditional cash transfer programs: Experimental evidence from Indonesia. *American Economic Journal: Economic Policy*, 12(4):88–110.

Sparrow, R., Suryahadi, A., and Widyanti, W. (2013). Social health insurance for the poor: Targeting and impact of Indonesia's Askeskin programme. *Social Science & Medicine*, 96:264–271.

Pisani, E., Kok, M. O., and Nugroho, K. (2017). Indonesia's road to universal health coverage: A political journey. *Health Policy and Planning*, 32(2):267–276.

Suryahadi, A., Widyanti, W., Perwira, D., and Sumarto, S. (2003). Minimum wage policy and its impact on employment in the urban formal sector. *Bulletin of Indonesian Economic Studies*, 39(1):29–50.

Del Carpio, X., Nguyen, H., and Wang, L. C. (2015). Does the minimum wage affect employment? Evidence from the manufacturing sector in Indonesia. *IZA Journal of Labor & Development*, 4:17.

VanderWeele, T. J. and Ding, P. (2017). Sensitivity analysis in observational research: Introducing the E-value. *Annals of Internal Medicine*, 167(4):268–274.

Sharma, A. and Kiciman, E. (2020). DoWhy: An end-to-end library for causal inference. *arXiv preprint arXiv:2011.04216*.

Battocchi, K., Dillon, E., Hei, M., Lewis, G., Lez, P., Li, Y., Maisog, J., and Oprescu, A. (2021). EconML: A Python Package for ML-Based Heterogeneous Treatment Effects Estimation. https://github.com/microsoft/EconML.

Brodersen, K. H., Gallusser, F., Koehler, J., Remy, N., and Scott, S. L. (2015). Inferring causal impact using Bayesian structural time-series models. *Annals of Applied Statistics*, 9(1):247–274.