
Testing Generalizability in Causal Inference

Anonymous Author
Anonymous Institution

Abstract

Ensuring robust model performance across diverse real-world scenarios requires addressing both transportability across domains with covariate shifts and extrapolation beyond observed data ranges. However, there is no formal procedure for statistically evaluating generalizability in machine learning algorithms, particularly in causal inference. Existing methods often rely on arbitrary metrics like AUC or MSE and focus predominantly on toy datasets, providing limited insights into real-world applicability. To address this gap, we propose a systematic and quantitative framework for evaluating model generalizability under covariate distribution shifts, specifically within causal inference settings. Our approach leverages the frugal parameterization, allowing for flexible simulations from fully and semi-synthetic benchmarks, offering comprehensive evaluations for both mean and distributional regression methods. By basing simulations on real data, our method ensures more realistic evaluations, which is often missing in current work relying on simplified datasets. Furthermore, using simulations and statistical testing, our framework is robust and avoids over-reliance on conventional metrics. Grounded in real-world data, it provides realistic insights into model performance, bridging the gap between synthetic evaluations and practical applications.

1 INTRODUCTION

Algorithm generalizability has garnered significant interest in fields such as computer vision and natural lan-

guage processing. It encompasses both transportability under covariate shifts between domains and extrapolation, where predictions are made within the same population but beyond the observed data range or in underrepresented subgroups.

Generalizability has also become a central focus in causal inference (Bareinboim and Pearl, 2016; Curth et al., 2021; Johansson et al., 2018; Buchanan et al., 2018; Ling et al., 2022; Bica and Schaar, 2022). Here, it refers to the ability of a causal model to make accurate causal predictions or draw valid causal conclusions when applied to data from a domain or distribution other than the one it was trained on. This concept is crucial when the objective involves understanding and predicting the effects of interventions across various settings. These settings may significantly diverge from the original conditions under which the model was developed, presenting challenges due to variations in factors such as environment, demographics, or other external influences. This holds particular importance in clinical settings, where the growing interest in personalized treatment and patient stratification underscores the need for inferences to generalize across diverse population domains.

Although strategies for improving generalization have been widely explored (Zhou et al., 2022; Yu et al., 2024), there has been comparatively little focus on developing a comprehensive, structured framework for evaluating generalizability. A common approach is to measure generalization or extrapolation performance using metrics like AUC for classification or MSE for regression. However, these metrics often lack informativeness. Achieving an MSE of 5, compared to 10 from other methods, on synthetic data irrelevant to the user’s intended application, does not provide clear guarantees regarding real-world performance. Therefore, it is essential to establish a systematic evaluation framework based on simulation for generalizability performance, which can offer a more robust and comprehensive understanding of how these methods perform on relevant tasks.

This paper proposes a method to statistically evaluate the generalizability of causal inference algorithms

under covariate and treatment distribution shifts. We introduce a semi-synthetic simulation framework using two domains – training (A) and testing (B) – that share the same intervened conditional outcome distributions but potentially differ in covariate and treatment distributions. A model is trained on domain A to **learn the shared high dimensional conditional outcome distribution**. We test the model’s generalizability by estimating the marginal causal quantities in domain B, where these values are explicitly known. This approach simplifies the evaluation process by reducing the complexity from higher-dimensional intervened models to a lower-dimensional causal effect, enabling more powerful statistical testing.

A high-level overview of the workflow of our method:

1. **Learn both the distribution parameters of two domains, and the Conditional Outcome Distribution (COD) from real-world data:** Define two domains, domain A and domain B, of which the covariate and treatment distributions can be different, but the COD is the same. These distributions can be learned empirically from real-world data, rather than just being limited to specifying parametric models.
2. **Model training:** Simulate semi-synthetic data from domain A using the distributions fitted on data. Train a conditional effect model on the simulated data.
3. **Prediction/Estimation:** Simulate data from domain B, whose covariate and treatment distributions may differ from domain A, but with an identical COD. Apply the trained model on the sampled covariates and treatments from domain B and estimate marginal causal quantities outcome predictions from the model.
4. **Evaluate generalizability with statistical testing:** Statistically test whether the estimated marginal causal quantities deviate significantly from the known ground truth in domain B. This provides an evaluation of the model’s generalizability under covariate and treatment distribution shifts. The tests assess whether the model generalizes effectively by focusing on lower-dimensional quantities instead of high-dimensional conditional outcome models.

Main Contributions In this work, we propose a formal framework for statistically testing the generalizability of machine learning algorithms under covariate and treatment distribution shifts, specifically in the context of causal inference. Rather than relying

on arbitrary metric scores, we provide tests that statistically evaluate the ability of both mean and distributional regression methods regarding generalizability. This provides a simple and effective solution for assessing how well algorithms account for these complexities in real-world applications.

Consequently, we claim that our evaluation method is:

- **Systematic:** We offer a structured framework that allows users to easily specify and input flexible data generation processes for simulations.
- **Comprehensive:** Our method supports simulations from various data generation processes, covering both continuous and discrete covariates and outcomes.
- **Robust:** We incorporate statistical testing to evaluate the generalizability of distributional and mean regression models.
- **Realistic:** Simulations can be based on actual data, bridging the gap between synthetic evaluations and real-world applications.

2 BACKGROUND

Throughout the paper, we consider a static treatment model with an outcome $Y \in \mathcal{Y} \subseteq \mathbb{R}$ and a general treatment X which can be either continuous or discrete. Let the set D of measured pretreatment covariates be $\mathbf{Z} \in \mathcal{Z} \subseteq \mathbb{R}^D$. If we make the standard assumptions of SUTVA, positivity, and conditional ignorability outlined in Pearl (2009), we define the marginal *causal* treatment density as follows:

$$p_{Y(x)}(y(x)) = \int p_{Y|\mathbf{Z}X}(y|\mathbf{z},x) p_{\mathbf{Z}}(\mathbf{z}) d\mathbf{z}, \quad (1)$$

which is the marginalized over the randomized model.

We also use $\mu(x) = \mathbb{E}[Y(X=x)]$ to denote the marginal potential outcome given the intervention. Correspondingly, we use $\mu(x,z) = \mathbb{E}[Y(X=x)|Z=z]$ to denote the conditional expectation of that potential outcome given covariate values. Note that $Y(x)$ is written as $Y|\text{do}(X=x)$ in the notation of Pearl (2009). When the treatment is binary, we define $\tau = \mathbb{E}[Y(1) - Y(0)]$ as the average treatment effect (ATE), quantifying the overall impact of a treatment or intervention across the entire population. Similarly, let $\tau(Z) = \mathbb{E}[Y(1) - Y(0)|Z]$ be the conditional average treatment effect (CATE), measuring the expected impact of an intervention for specific subgroups or individuals, capturing treatment effect heterogeneity.

We aim to evaluate the generalizability of an outcome regression model $\hat{f}(X, Z)$ that predicts the expected outcome Y , with the model’s predicted outcomes indicated by a hat symbol.

2.1 Generalizability in Causal Inference

Extensive research has focused on generalizability in causal inference, as mentioned in the Introduction. As highlighted by Ling et al. (2022), three common approaches are used to assess treatment effect generalizability: inverse probability of sampling weighting (IPSW) methods that adjust for differences between study and target populations by weighting based on sample inclusion probabilities (Buchanan et al., 2018); outcome model-based methods that estimate the conditional outcome directly (Kern et al., 2016); and the hybrid approaches that combines both (Dahabreh et al., 2019).

In this work, we focus on algorithms that generalize outcome predictions across different domains, enabling accurate CATE or COD estimation. This is crucial for understanding individual-level treatment effect heterogeneity and ensuring models can adapt to new populations or environments with varying covariate distributions. A summary of common CATE estimation methods is provided by Caron et al. (2022).

Despite advancements in CATE estimation, a systematic framework for evaluating generalizability is still underdeveloped. Current evaluation methods, like MSE and Precision in Estimation of Heterogeneous Effect (PEHE), provide limited real-world insights (Curth et al., 2021; Kiriakidou and Diou, 2022). To address this gap, we propose a systematic approach to evaluate how well CATE algorithms perform across domains with different covariate distributions, offering a more practical assessment of generalizability.

2.2 Frugal Parameterization

A frugal parameterization of an observational joint distribution, $P_{\mathbf{Z}XY}$, factorizes the distribution into a set of causally relevant components (Evans and Didelez, 2024). This decomposition explicitly parameterizes the marginal causal effect, $P_{Y(x)}$ and builds the rest of the model around it.

Let us start by first parameterizing the *conditional outcome distribution* (COD), $P_{Y(x)|\mathbf{Z}}$. Frugal models parameterize the COD in terms of the marginal causal effect, $P_{Y(x)}$, and a conditional copula distribution, $C_{Y(x)|\mathbf{Z}}$. Here, $C_{Y(x)|\mathbf{Z}}$ models the joint dependency between the marginal causal distribution and each of the univariate marginal covariate distributions, $\{P_{Z_i}\}_i$

such that:

$$P_{Y(x)|\mathbf{Z}} = P_{Y(x)} \cdot C_{Y(x)|\mathbf{Z}}, \quad (2)$$

where $C_{Y(x)|\mathbf{Z}}$ is a copula distribution function (see Supplementary Material for further details on copulas) on the ranks of the marginal probability integral transform of the covariates:

$$C_{Y(x)|\mathbf{Z}} := C(F_{Y(x)} | F_{Z_1}, \dots, F_{Z_D}). \quad (3)$$

This leaves the distribution of the *past*, $P_{\mathbf{Z}X}$, i.e. the covariate distribution and the propensity score. Note that we assume that all covariates are strictly pretreatment, i.e. \mathbf{Z} cannot include any mediators. The past and the COD are variation independent, in the sense that they parameterize separate, non-overlapping aspects of the joint distribution (Evans and Didelez, 2024). This allows the past to be freely specified without affecting the conditional copula, nor the marginal causal effect.

3 METHOD

Figure 1 provides an overview of our workflow. We begin by defining both a test and a training domain, each with a distribution over the pretreatment covariates and the treatment, allowing for distribution shifts across covariates and treatment allocation. The COD is frugally parameterized with a conditional copula, where the covariates’ cumulative distribution functions (CDFs) are derived from the test domain’s covariate densities. This ensures that samples from the test dataset follow a **known, customizable** marginal causal density, $p_{Y(x)}$.

The training data is generated from the same COD but with a non-analytic marginal causal density, as the training covariate densities do not match the covariate CDFs used to parameterize the conditional copula. We then train a model, $\hat{f}(x, \mathbf{z})$, on the training data. Finally, a statistical test is performed to validate whether the lower dimensional marginal quantity (e.g. ATE, τ , or the expected potential outcome, $\mu(x)$) estimated using model outcomes equals the ground truth in the test domain.

3.1 Data Simulation

In this section we discuss how we can simulate the data. We show how we can construct two datasets with covariate/treatment domain shift with the exact same COD, but where in one domain the marginal causal effect is well understood.

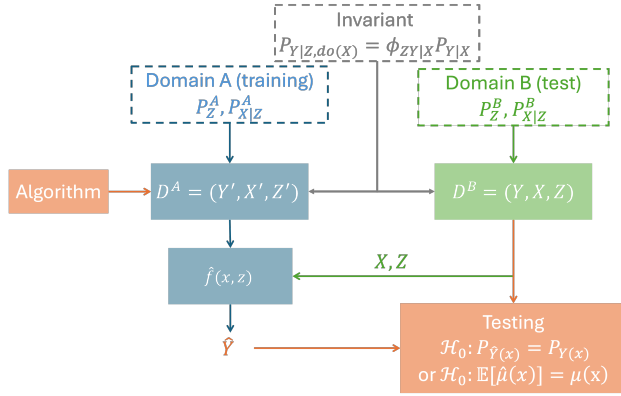


Figure 1: Workflow of the Proposed Method.

3.1.1 Multi-domain Simulation with Frugal Models

We begin by specifying two data generating processes: the training data, $D^A \sim P_{\mathbf{Z}XY}^A$, and the test data, $D^B \sim P_{\mathbf{Z}XY}^B$. Our goal is to construct a COD that parameterizes the joint density across both domains, while ensuring that the marginal causal density in domain B is parameterized by $p_{Y(x)}$.

Recall from Section 2.2 that a general observational density can be factorized into the *past*, $p_{\mathbf{Z}X}$, and the COD, $p_{Y(x)|\mathbf{Z}}$:

$$p_{Y(x)|\mathbf{Z}}(y | \mathbf{z}) = p_{Y(x)}(y) \times c_{Y(x)|\mathbf{Z}}(F_{Y(x)}(y) | F_{Z_1}(z_1), \dots, F_{Z_D}(z_D)) \quad (4)$$

where $F_{Y(x)}$ is the CDF associated with the marginal causal density $p_{Y(x)}$.

Note that the copula density in Equation (4) is not only determined by the copula's family and its parameterization, but also by the choice of marginal CDFs for the covariates, \mathbf{Z} . If the conditional copula density is marginalized over the densities corresponding to the covariate CDFs, then the ranks of the marginal causal density will be uniformly distributed:

$$p(F_{Y(x)}) = \int d\mathbf{z} \, c_{Y(x)|\mathbf{Z}}(y(x) | \mathbf{z}) \cdot \prod_{d=1}^D p_{Z_d}(z_d) = 1. \quad (5)$$

However, this uniformity is guaranteed only if the marginal covariate densities $\{p_{Z_d}\}_{d=1}^D$ correspond to the CDFs used to parameterize the copula. If we instead consider a set of alternative marginal densities, $\{p'_{Z_d}\}_{d=1}^D$, are not derived from the CDFs within the copula, i.e. $F_{Z_d}(Z_d = t) \neq F_{Z'_d}(Z'_d = t)$ then the

rank uniformity is not assured. Thus, the COD density is generally valid under any distribution of the past, and will not guarantee the sampling from the specified marginal causal density if the covariate densities are derived from the CDFs that parameterize the copula. We present the conditions by which alternative distributions will yield samples drawn from the specified marginal causal density, assuming that the conditional copula density is Gaussian in the Supplementary Material. Given how rarely these conditions are satisfied, we do not believe this will commonly be encountered in semi-synthetic benchmark generation. These conditions will likely be even harder to satisfy if a more complex multivariate copula (such as non-Gaussian vine) is chosen.

For evaluating generalization, we set the CDFs within the copula density to be derived from the covariate densities in the test domain $P_{\mathbf{Z}XY}^B$. This allows us to construct the COD density across all covariate spaces,

$$p_{Y(x)|\mathbf{Z}}(y | \mathbf{z}) = p_{Y(x)}^B(y) \times c_{Y(x)|\mathbf{Z}}(F_{Y(x)}^B(y) | F_{Z_1}^B(z_1), \dots, F_{Z_D}^B(z_D)) \quad (6)$$

which will sample from a known marginal causal density equal to $p_{Y(x)}$ if the covariate CDFs in the copula are derived from the test domain covariate densities.

This offers a great deal of flexibility in testing method generalizability. One can draw training and test datasets with different covariate densities and propensity scores, while guaranteeing that the CODs remain consistent, and that the test data is drawn from a distribution with a marginal causal density parameterized by $p_{Y(x)}$.

3.1.2 Generating Semi-Synthetic Benchmarks

In real-data based simulations, we follow the workflow outlined in Algorithm 1. First, we estimate the empirical CDFs of the pretreatment covariates for the test data, denoted as $\hat{F}_{Z_d}^B$, $\forall d = \{1, \dots, D\}$. We then estimate the marginal causal density $\hat{p}_{Y(x)}^B$ and the joint copula $\hat{c}_{Y(x)|\mathbf{Z}}^B$, capturing the covariate-outcome dependency conditional on treatment. With the test copula known, we draw samples $\mathbf{u}_{\mathbf{Z}}^B \sim \hat{c}_{Y(x)|\mathbf{Z}}^B$, and use inverse transforms to generate the covariate samples $z_d^B = \hat{F}_{Z_d}^{B^{-1}}(u_{Z_d}^B)$. Next, we estimate the propensity score model for the test data, $\hat{p}_{X|\mathbf{Z}}^B$ and sample the treatment variable $x^B \sim \hat{p}_{X|\mathbf{Z}}^B(\cdot | \mathbf{z}^B)$. The outcome for the test data calculating using $y^B = \hat{F}_{Y(x)}^{B^{-1}}(u_{Y(x)}^B)$, where $u_{Y(x)}^B$ is the sampled outcome rank from the copula. For the training data, we follow a similar approach. Details can be found in Algorithm 1.

Algorithm 1 Semi-synthetic Data Generation.

Input: Original test data; original covariates and treatment from training data.

Parameter estimations on test domain

Estimate test empirical CDFs, $\{\hat{F}_{Z_d}^B\}_{d=1}^D$; marginal causal density, $\hat{p}_{Y(x)}^B$; joint copula, $\hat{c}_{ZY(x)}^B$; propensity score model $\hat{p}_{X|Z}^B$.

Transformation on test domain

Sample $(\mathbf{u}_Z^B, u_{Y(x)}^B) \sim \hat{c}_{ZY(x)}^B$;

Calculate $\{z_d^B = \hat{F}_{Z_d}^{B-1}(u_{Z_d}^B)\}_{d=1}^D$;

Sample $x^B \sim \hat{p}_{X|Z}^B(\cdot | \mathbf{Z}^B)$;

Calculate $y^B = \hat{F}_{Y(x)}^{B-1}(u_{Y(x)}^B)$.

Parameter estimation on training domain

Estimate training empirical CDFs, $\{\hat{F}_{Z_d}^A\}_{d=1}^D$; covariate copula, \hat{c}_Z^A ; propensity score model, $\hat{p}_{X|Z}^A$.

Transformation on training domain

Sample $u_Z^A \sim \hat{c}_Z^A$;

Calculate $\{z_d^A = \hat{F}_{Z_d}^{A-1}(u_{Z_d}^A)\}_{d=1}^D$;

Sample $x^A \sim \hat{p}_{X|Z}^A(\cdot | \mathbf{z}^A)$;

Sample $u_{Y(x)}^A \sim \hat{c}_{Y(x)|Z}^B(\cdot | \hat{F}_{Z_1}^B(z_1^A), \dots, \hat{F}_{Z_D}^B(z_D^A))$;

Calculate $y^A = \hat{F}_{Y(x)}^{B-1}(u_{Y(x)}^A)$.

Output: Training sample $D^A = (\mathbf{z}^A, x^A, y^A)$;
Test sample $D^B = (\mathbf{z}^B, x^B, y^B)$.

3.2 Statistical Testing

Given that we know the marginal causal density parameterized by $p_{Y(x)}$ from the frugal parameterization, we are able to develop the statistical testing on $\mathcal{H}_0 : \mathbb{E}[\hat{\mu}(x)] = \mu(x)$ instead of $\mathcal{H}_0 : \mathbb{E}[\hat{\mu}(x, \mathbf{z})] = \mu(x, \mathbf{z})$ for mean regression models, and $\mathcal{H}_0 : \hat{P}_{Y(x)} = P_{Y(x)}$ instead of $\mathcal{H}_0 : \hat{P}_{Y(x)|Z} = P_{Y(x)|Z}$ for distributional regression.

Our testing algorithms require some parameters: N_B as the number of bootstrap samples, N^{tr} , N^{te} as the number of samples simulated from training domain and test domain for each bootstrap, respectively; for distributional testing, we also need to specify N_Y , which is the number of outcome samples simulated from distributional regression output for each $\hat{f}(x, \mathbf{z})$. We provide testing methods for two types of regression models: mean regression in Algorithm 2 or distributional regression in Algorithm 3. Note that, in Algorithm 2, we can replace $\mu^{te}(x)$ with τ^{te} as the reference target when X is binary, which is what we used in our experiments. The testing method used in Algorithm 3 can also be replaced by other statistical tests, e.g. Maximum Mean Discrepancy Test (Gretton et al., 2012) or the Cramér-von Mises Test (Anderson, 1962).

Algorithm 2 Generalizability Evaluation on Mean Regression Models.

Input: Θ^{tr} : parameters for training domain,
 Θ^{te} : parameters for test domain,
 $\mu^{te}(x^0)$: reference.

for $b = 1, \dots, N_B$ **do**

Draw $D_b^{tr} := \{(\mathbf{z}'_{ib}, x'_{ib}, y'_{ib})\}_{i=1}^{N^{tr}} \sim P_{\Theta^{tr}}$;

Fit the mean regression model, \hat{f} , on D_b^{tr} ;

Draw $D_b^{te} := \{(\mathbf{z}_{ib}, x_{ib})\}_{i=1}^{N^{te}} \sim P_{\Theta^{te}}$;

Apply \hat{f} on D_b^{te} to get predictions $\{\hat{f}(x_{ib}, \mathbf{z}_{ib})\}_{i=1}^{N^{te}}$;

Calculate

$$\hat{\mu}_b^{te}(x^0) = \frac{\sum_{i=1}^{N^{te}} \mathbb{1}(x_{ib} = x^0) \hat{f}(x_{ib}, \mathbf{z}_{ib})}{\sum_{i=1}^{N^{te}} \mathbb{1}(x_{ib} = x^0)}.$$

end for

Get the p-value p by conducting a t-test to compare the target parameter $\mu^{te}(x^0)$ and the distribution of $\{\hat{\mu}_b^{te}(x^0)\}_{b=1}^{N_B}$.

Return p .

A summary of this workflow is presented in Figure 1.

4 EXPERIMENTS

In this section, we use our workflow to evaluate the generalizability of a range of modern causal models.

As discussed in several review papers like Curth et al. (2021), Ling et al. (2022) and Kiriakidou and Diou (2022), methods such as Meta-Learners (e.g. T- and S-learners) (Künzel et al., 2019), CausalForest (Wager and Athey, 2018), TARNet (Shalit et al., 2017), and BART (Chipman et al., 2010) are widely used for CATE estimation, each offering advantages in different scenarios. Our evaluation focuses on their performance under covariate distribution shifts, specifically examining the accuracy of their CATE estimations. Further details can be found in the Supplementary Material.

Another interesting algorithm to be evaluated is engression, introduced in Shen and Meinshausen (2023). It approximates the conditional distribution using a pre-additive noise model. Targeting at a distributional regression, the model is capable of extrapolating to unseen or underrepresented data points through its learned non-linear transformations. The key factors which affect engression’s generalizability are the distances between two domains, and whether the true underlying function must be strictly monotonic in the extrapolation region. In our experiments, we evaluate engression in both the S-learner and T-learner settings.

Algorithm 3 Generalizability Evaluation on Distributional Regression Models.

Input: Θ^{tr} : parameters for training domain,
 Θ^{te} : parameters for test domain,
 $P_{Y(x^0)}^{te}$: reference.

for $b = 1, \dots, N_B$ **do**
 Sample $D_b^{tr} := \{(z'_{ib}, x'_{ib}, y'_{ib})\}_{i=1}^{N^{tr}} \sim P_{\Theta^{tr}}$;
 Fit the distributional regression model, \hat{f} , on D_b^{tr} ;
 Sample $D_b^{te} := \{z_{ib}, x_{ib}\}_{i=1}^{N^{te}} \sim P_{\Theta^{te}}$;
 Apply \hat{f} on D_b^{te} to get distributional predictions $\hat{P}_{Y(x_{ib})|z_{ib}}$;
 For each i , sample $\{y_{ib}^j\}_{j=1}^{N_Y}$ from $\hat{P}_{Y(x_{ib})|z_{ib}}$.
end for
 Estimate $\hat{P}_{Y(x^0)}^{te} = \bigcup_{b=1}^{N_B} \bigcup_{i=1}^{N^{te}} \bigcup_{j=1}^{N_Y} \{y_{ib}^j \mid x_{ib} = x^0\}$.
 Conduct distribution tests, e.g. the Kolmogorov-Smirnov test, for $\mathcal{H}_0 : \hat{P}_{Y(x^0)}^{te} = P_{Y(x^0)}^{te}$ and get the p-value p .
Return p .

4.1 Synthetic Data

We first conduct experiments on synthetic data to demonstrate and validate our method. While our approach can handle various data types and is particularly effective with high-dimensional covariates and continuous treatment interventions, for clarity, in this simple example, we focus on two continuous confounders, Z_1 and Z_2 , sampled from identical gamma distributions, with a binary intervention X . We first focus on a randomized controlled trials (RCT) setting, $X \sim \text{Bernoulli}(0.5)$. Note that these parameters can be different between the two domains; here we just make them identical for simplicity in this experiment. We parameterize the Gaussian copula, $c_{ZY(X)}$, with Spearman correlation coefficients $\rho_{Z_1 Z_2} = 0$, $\rho_{Z_1 Y(X)} = 0.1$ and $\rho_{Z_2 Y(X)} = 0.9$. The causal effect, $P_{Y(X)}^{te}$ is defined as $\mathcal{N}(1 + 2X, 1)$ in the test domain. For the simulation, we generate $N^{tr} = 200$ training samples and $N^{te} = 50$ test samples per bootstrap, with $N_B = 200$ bootstraps in total, repeating this process for 50 iterations. The marginal distributions of Z_1 and Z_2 in training domain follow identical Gamma distributions with shape $k = 1$ and rate $\theta = 1$.

We examine two settings: in Setting 1, the test domain has a slight covariate shift, with Z_1 and Z_2 following a Gamma distribution of $k = 2$, $\theta = 1$. In Setting 2, the shift is more significant ($k = 4$, $\theta = 1$). Despite these shifts, the COD remains the same due to frugal parameterization, as shown in Figure 2.

The p-values in Figure 3 illustrate the differences

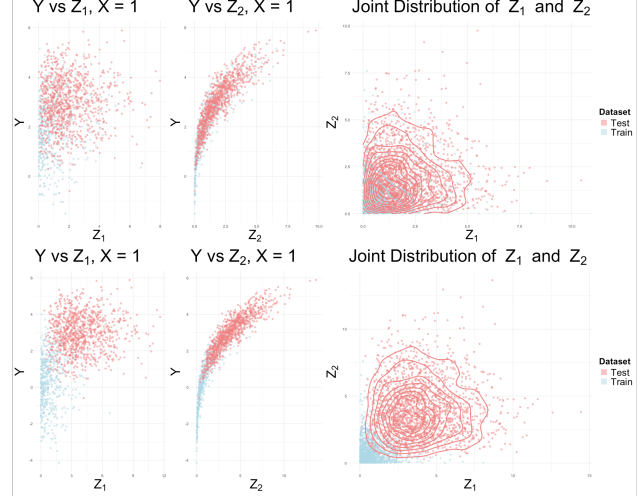


Figure 2: Synthetic Data Generated from Setting 1 (Top) and Setting 2 (Bottom).

across models. As expected, with a more significant domain shift in Setting 2, models face greater difficulty in generalizing, as reflected by the smaller p-values generally compared to Setting 1. T-BART and T-engression showed good generalizability performances in this specific setting. TARNet struggles, likely due to the complexity of its representation learning network design and hyperparameter tuning.

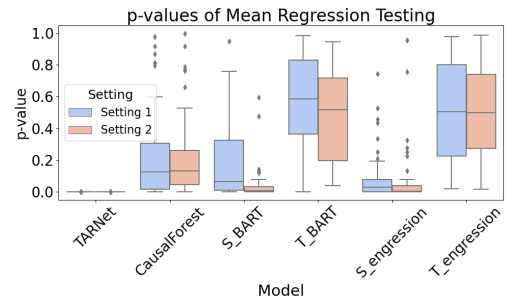


Figure 3: p -values of Mean Regression Testing, Synthetic Data of 50 Iterations.

With our method, we are able to test the generalizability of distributional regression. Figure 4 demonstrates the p-values of distributional regression testing of S-engression under the two settings, with $N_Y = 50$. Not surprisingly, since the covariate distribution shift in Setting 1 is smaller, S-engression demonstrates good

generalizability compared to that in Setting 2.

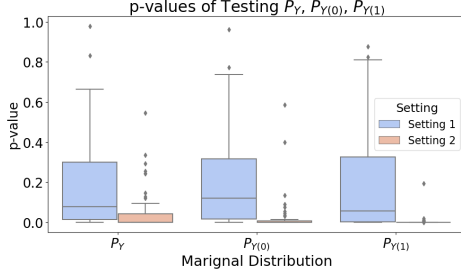


Figure 4: p -values of Distributional Regression Testing (Kolmogorov–Smirnov Test) of S-engression, Synthetic Data of 50 Iterations.

Supported by flexible simulations based on actual data, our method is useful for stress testing and model diagnostics. Figure 5 illustrates an example where we examine how varying the training set size affects the generalizability of T-BART and T-engression. The generalizability performances of T-BART and T-engression worsen as N^{tr} exceeds 100. This issue may stem from problems like overfitting, but solving these problems is not our focus. Rather, our method serves as a tool to detect and highlight potential issues when making predictions on real data, which is feasible with the simulation based on actual data using the frugal parameterization.

Note that extrapolation performance for models like engression is typically evaluated visually, one dimension at a time. Our method, however, offers significant advantages by providing statistical evaluation of extrapolation performance in high-dimensional covariates.

4.2 Real Data

We evaluate algorithm generalizability using the Infant Health and Development Program (IHDP) dataset, a randomized experiment conducted between 1985 and 1988 to study the effect of home visits on infants’ cognitive test scores (Hill, 2011). This dataset has become widely used in domain adaptation research (Johansson et al., 2018; Curth et al., 2021; Shi et al., 2021).

The IHDP dataset contains $T = 1000$ trials, each consisting of the same 747 subjects and 25 covariates, with the first six being continuous and the rest binary. The potential outcomes $Y(1)$ and $Y(0)$ are provided in the data. In each trial t , $Y(0) \sim \mathcal{N}(\mathbf{Z}\beta_t, 1)$, $Y(1) \sim \mathcal{N}(\mathbf{Z}\beta_t + 4, 1)$, and β_t is randomly chosen from a set of vectors. Thus, the potential outcomes vary

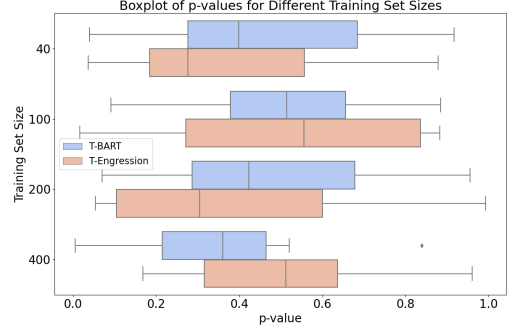


Figure 5: p -values of Mean Regression Testing of 50 Iterations, Varying N^{tr} , Setting 2, Synthetic Data.

across trials, while the covariates, CATE and ATE remain constant.

We first demonstrate algorithm generalizability by treating both domains as RCTs, i.e. setting the propensity score model as $X \sim \text{Bernoulli}(0.5)$ for all units. The observed outcome is then $Y = XY(1) + (1 - X)Y(0)$. We randomly select 50 trials from the 1000 available, with each trial used to create one training-test pair, and evaluate the model’s generalizability on them. To introduce domain shift, we keep all covariate values identical between the training and test domains, except for Z_1 , which is set to 1.5 times the original value in the test domain compared to the training domain. For each training-test pair, we learn the parameters following Algorithm 1, specifying the marginal causal distribution to follow a Gamma distribution. We denote the resulting data generation distributions as $P_{\Theta^{tr}}$, $P_{\Theta^{te}}$ for the training and test domains, respectively. We sample training data of $N^{tr} = 1000$ from $P_{\Theta^{tr}}$, and $N^{te} = 200$ test data from $P_{\Theta^{te}}$. The number of bootstraps is set to be $N_B = 200$.

Figure 7 shows the boxplot of p -values of each model and Table 1 contains the percentage of p values greater than 0.05 across the 50 trials. T-/S-engression demonstrate better generalizability in this setting among all these methods. We also give the result of distributional regression testing in Figure 8.

While we use the RCT setting as an example above to demonstrate our method, it is also applicable to observational studies. The percentage of $p > 0.05$ across 50 trials of each algorithm, when treatment arms are imbalanced in each trial by setting $P(X = 1 | Z) = \text{logit}(Z_2 + Z_3 + Z_4)$ can be found in Table 1. Since our paper’s focus is on providing a systematic generalizability evaluation method, we omit further analysis

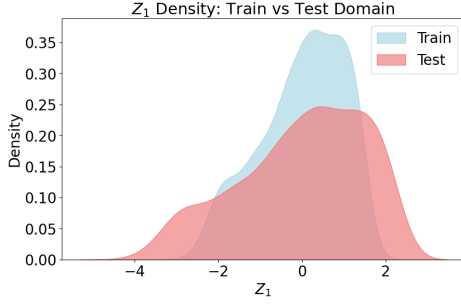


Figure 6: Density of Z_1 of Training and Test Domains.

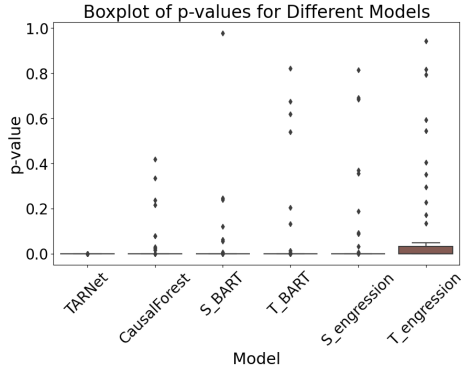


Figure 7: p -values of Mean Regression Testing of 50 Trials in IHDP.

here.

Details on hyperparameters and additional experiments, including performance comparisons with or without domain shift when the CATE is known to be linear, are provided in the Supplementary Material.

5 SUMMARY

In this paper, we develop a statistical method for evaluating the generalizability of causal inference algorithms using actual application data, facilitated by frugal parameterization. Our approach introduces a semi-synthetic simulation framework that bridges the gap between synthetic simulations and real-world applications, supporting the generalizability evaluation of both mean and distributional regression models. Through flexible, user-defined data generation processes, our framework provides robust statistical testing to assess how well models trained in one domain

Table 1: Percentage of $p > 0.05$, across 50 Trials.

Model	RCT	Non-RCT
TARNet	0	0
CausalForest	12%	6%
S-BART	12%	8%
T-BART	12%	6%
S-engression	18%	6%
T-engression	24%	8%

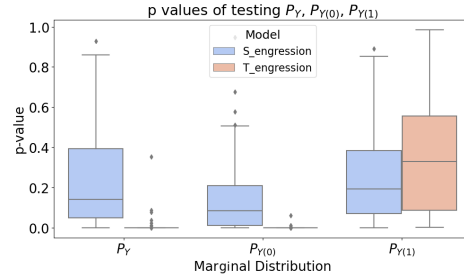


Figure 8: p -values of Distributional Regression Testing of 50 Trials in IHDP.

generalize to shifted domains.

Through experiments on the synthetic and IHDP datasets, we assess the generalizability of algorithms such as TARNet, CausalForest, S-/T-BART, S-/T-engression under domain shift. Our method acts as a valuable diagnostic tool, allowing us to explore how factors like training set size or covariate shifts impact generalizability. These insights can help identify model strengths and weaknesses and inform how causal inference models adapt to different settings.

We remark that our approach of rejecting the null hypothesis shows that a model is not generalizable, but it does not quantify the extent of failure. An extension of this approach may be to develop a more flexible testing method, inspired by equivalence testing (Wellek, 2002). This would assess not just whether a model fails but also by how much, determining if its performance is significantly worse than a given threshold. This offers a more nuanced view than traditional hypothesis testing. In this paper, we only consider marginal causal quantities as the validation references, but our framework can be easily adapted to use lower dimensional CODs as the reference instead.

We hope that this work inspires a more careful consideration of model evaluation, encourages simulations that better reflect real-world conditions, and highlights

the importance of stress testing in advancing causal inference methodologies.

References

- Anderson, T. W. (1962). On the distribution of the two-sample cramer-von mises criterion. *The Annals of Mathematical Statistics*, pages 1148–1159.
- Bareinboim, E. and Pearl, J. (2016). Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113:7345–7352.
- Bica, I. and Schaar, M. (2022). Transfer learning on heterogeneous feature spaces for treatment effects estimation. In *Advances in Neural Information Processing Systems*, volume 35, pages 37184–37198.
- Buchanan, A. L., Hudgens, M. G., Cole, S. R., Molan, K. R., Sax, P. E., Daar, E. S., Adimora, A. A., Eron, J. J., and Mugavero, M. J. (2018). Generalizing evidence from randomized trials using inverse probability of sampling weights. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 181(4):1193–1209.
- Caron, A., Baio, G., and Manolopoulou, I. (2022). Estimating individual treatment effects using non-parametric regression models: A review. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 185(3):1115–1149.
- Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). BART: Bayesian additive regression trees.
- Curth, A., Svensson, D., Weatherall, J., and Schaar, M. (2021). Really doing great at estimating CATE? a critical look at ML benchmarking practices in treatment effect estimation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Dahabreh, I. J., Robertson, S. E., Tchetgen, E. J., Stuart, E. A., and Hernán, M. A. (2019). Generalizing causal inferences from individuals in randomized trials to all trial-eligible individuals. *Biometrics*, 75(2):685–694.
- Evans, R. and Didelez, V. (2024). Parameterizing and simulating from causal models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86:535–568.
- Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.
- Hill, J. L. (2011). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240.
- Johansson, F., Kallus, N., Shalit, U., and Sontag, D. (2018). Learning weighted representations for generalization across designs. *arXiv preprint arXiv:1802.08598*.
- Kern, H. L., Stuart, E. A., Hill, J., and Green, D. P. (2016). Assessing methods for generalizing experimental impact estimates to target populations. *Journal of Research on Educational Effectiveness*, 9(1):103–127.
- Kiriakidou, N. and Diou, C. (2022). An evaluation framework for comparing causal inference models. In *Proceedings of the 12th Hellenic Conference on Artificial Intelligence*, pages 1–9.
- 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.
- Ling, A., Montez-Rath, M., Carita, P., Chandross, K., Lucats, L., Meng, Z., Sebastien, B., Kapphahn, K., and Desai, M. (2022). A critical review of methods for real-world applications to generalize or transport clinical trial findings to target populations of interest. *arXiv preprint arXiv:2202.00820*.
- Pearl, J. (2009). *Causality*. Cambridge University Press.
- Shalit, U., Johansson, F. D., and Sontag, D. (2017). Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pages 3076–3085. PMLR.
- Shen, X. and Meinshausen, N. (2023). Engression: Extrapolation for nonlinear regression? *arXiv preprint arXiv:2307.00835*.
- Shi, C., Veitch, V., and Blei, D. (2021). Invariant representation learning for treatment effect estimation. In *Uncertainty in Artificial Intelligence*, pages 1546–1555.
- 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.
- Wellek, S. (2002). *Testing statistical hypotheses of equivalence*. Chapman and Hall/CRC.
- Yu, H., Liu, J., Zhang, X., Wu, J., and Cui, P. (2024). A survey on evaluation of out-of-distribution generalization. *arXiv preprint arXiv:2403.01874*.
- Zhou, K., Liu, Z., Qiao, Y., Xiang, T., and Loy, C. C. (2022). Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:4396–4415.

Checklist

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] *We do our utmost to make this clear in our submission.*
 - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] *We are explicit about the sample sizes used in the paper, and have no inference algorithms as such to report.*
 - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] *We attach a requirements file to our submitted code.*
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. [Yes] *We make this clear in either the main body or the Supplementary Material.*
 - (b) Complete proofs of all theoretical results. [Yes] *Relevant proofs are either referenced or added to the Supplementary Material.*
 - (c) Clear explanations of any assumptions. [Yes] *We tried our best to make them clear.*
3. For all figures and tables that present empirical results, check if you include:
 - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] *All relevant code is included in our attached code. All external data we use is cited.*
 - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] *We discuss our fitting process in the Supplementary Materials.*
 - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] *Done.*
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] *We discuss computational requirements.*
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
 - (a) Citations of the creator If your work uses existing assets. [Yes/No/Not Applicable] *Cited in Supplementary Material.*
 - (b) The license information of the assets, if applicable. [Yes]
 - (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
 - (d) Information about consent from data providers/curators. [Yes]
 - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable] *We don't use sensitive material.*
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
 - (a) The full text of instructions given to participants and screenshots. [Not Applicable] *No crowdsourcing or human subjects used.*
 - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable] *No crowdsourcing or human subjects used.*
 - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable] *No crowdsourcing or human subjects used.*