Causal Effect Estimation via Smooth Counterfactual Trajectories

Dylan Geldenhuys

Deep Learning IndabaX 2025

Introduction & Motivation

In telecom networks, improving one KPI (like RSRQ) often triggers changes in others (e.g. throughput, utilization), which jointly influence customer experience and Net Promoter Score (NPS).

Yet most causal ML models simulate interventions by perturbing a single variable in isolation — ignoring mediated effects and assuming all counterfactuals are supported by the data.

This creates major limitations for industry-scale decision-making:

- **Limited Scalability** – A/B tests or pilots are narrow in scope, rarely generalizing across regions, devices, or network loads.
- **Confounded Insights** – Multiple initiatives (e.g. promos + upgrades) often run together, making it hard to isolate what truly drives outcomes.
- **Lack of Personalization** – Generic uplift models overlook variation by device, location, or usage, missing targeted high-impact opportunities.

We propose a practical causal simulation framework that:

- Simulates mediated counterfactuals by propagating downstream effects;
- Validates overlap using a smoothness-based check on covariate continuity;
- Estimates CATE with an S-learner trained on factual + valid simulations;
- And reweights samples by local coverage, improving generalization in low-support regions.

This creates a foundation for a scalable, interpretable causal digital twin that can simulate interventions across the entire network — enabling personalized, ROI-aware decisions.

Methodology: Overlap-Validated Mediated Simulation Pipeline

Step 1. Perturb the Treatment

Simulate a hypothetical intervention, e.g., increasing RSRQ by +1 dB:

$$T' = T + \Delta$$

Step 2. Propagate Mediated Effects

Estimate downstream changes (e.g., DLTP, utilization, CQI) using:

$$X' = \mathcal{F}_{\text{med}}(X, T')$$

$\mathcal{F}_{\text{med}}$ is implemented using **K-nearest neighbors (KNN)**, selecting similar users who experienced the perturbed RSRQ and averaging their observed KPIs:

$$\mathbf{M}' = \mathbb{E}_{x_j \in \mathcal{N}(x,T')}[M_j]$$

Step 3. Validate Counterfactual Coverage

We quantify the support and continuity of the counterfactual transition $\mathbf{x} \to \mathbf{x}'$ via a smoothness score $S \in (0, 1]$.

Interpolated path:

$$\mathbf{x}_\alpha = (1 - \alpha)\mathbf{x} + \alpha\mathbf{x}', \quad \alpha \in [0, 1]$$

Local instability:

$$\iota(\mathbf{x}_\alpha) = \mathbb{E}_{\mathbf{z} \in \mathcal{N}(\mathbf{x}_\alpha)} \left[\frac{\sigma_f}{\mu_f + \epsilon} \right]$$

Average path instability:

$$I(\mathbf{x}, \mathbf{x}') = \frac{1}{K} \sum_{i=1}^{K} \iota(\mathbf{x}_{\alpha_i})$$

Smoothness score:

$$S(\mathbf{x}, \mathbf{x}') = \exp(-I(\mathbf{x}, \mathbf{x}'))$$

Interpretation: Higher S implies smoother, better-supported transitions; low S flags extrapolation or coverage gaps.

Step 4. Apply Coverage-Based Reweighting

We weight each training sample by its local covariate density $\rho(X)$:

- $w_i \propto \rho(X_i)$
- Downweight sparse regions, upweight well-covered ones

Step 5. Train a Causal S-Learner

Train a model on factual + valid counterfactual data:

$$\hat{\mu}(X, T) \approx \mathbb{E}[Y \mid X, T] \quad \Rightarrow \quad \hat{\tau}(X) = \hat{\mu}(X, 1) - \hat{\mu}(X, 0)$$

We used **XGBoost**; other options include random forests or neural networks.

Method overview

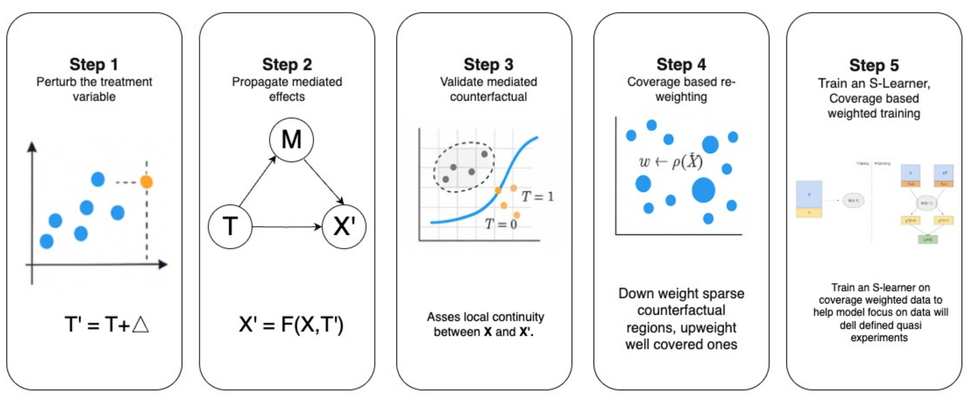

Figure: Illustration of applied steps.

Preliminary experiments

Data: We used a custom-built simulation of telecom sessions across cell towers. KPIs such as RSRQ, DLTP, PRB utilization, and CQI were generated per session. Users were placed geographically and moved between towers based on spatial trajectories.

Step 1 – User journeys

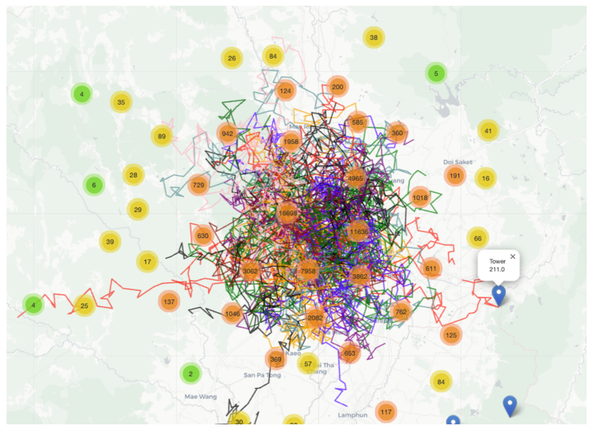

Figure: Simulated users distributed across towers.

Step 2 – Counterfactual Generation

We incrementally perturbed RSRQ and applied KNN to simulate changes in downstream KPIs (DLTP, PRB utilization, CQI). Smoothness of KPI transitions was used to validate the counterfactuals.

Smoothness from factual to counterfactual co-variates

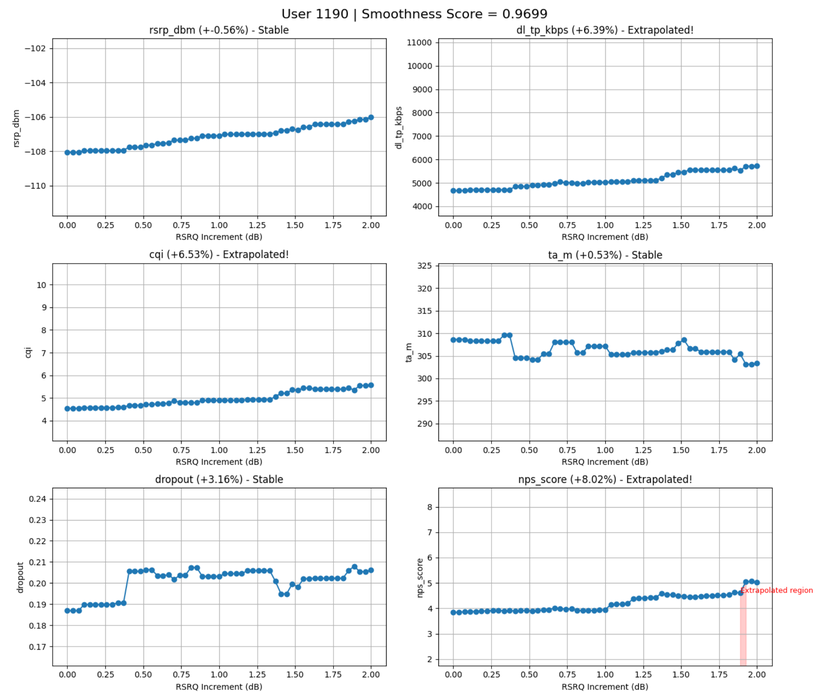

Figure: Correlation between predicted and true CATE for weighted model.

Step 3 – Model Evaluation

We trained two S-learner models:

- **Unweighted**: standard training on valid samples
- **Weighted**: trained with local coverage weights

Performance Table:

Model	PEHE	ATE Error	RMSE
S-Learner (No mediation)	6.4	6.3	0.145
S-Learner (unweighted)	0.165	0.019	0.139
S-Learner (weighted)	**0.154**	**0.018**	**0.145**

Discussion & Conclusion

We present a practical framework for causal effect estimation that improves generalization by filtering and reweighting based on local covariate coverage. Our approach consists of three key components:

- **Trajectory-Based Simulation:** We perturb treatment (e.g., RSRQ) and simulate realistic counterfactuals using KNN-based feature propagation.
- **Smoothness-Based Coverage Scoring:** For each simulated path, we compute a local instability score that quantifies how stable the transition is across key features. This identifies unsupported or extrapolated regions.
- **Coverage-Weighted Learning:** These scores are used to weight samples in a standard S-learner, emphasizing well-supported examples and down-weighting noisy regions.

We find that coverage-weighted models consistently outperform baselines in PEHE, ATE error, and predictive accuracy. Improvements are robust across treatment types and hold under realistic distribution shifts.

This approach is particularly well-suited to industry settings with observational data and unobserved counterfactuals. It supports interpretable, modular digital twins for intervention planning — for example, optimizing customer experience in telecom networks.

Key Contribution: We treat positivity as a continuous, measurable signal — derived from smoothness along counterfactual paths — and use it to guide model training. This bridges causal theory and applied decision-making.