Empirical Study on the Impact of Network Architecture on Causal Effect Estimation with TARNet

Monika Witczak[1]
Supervisor(s): Jesse Krijthe[1]**, Rickard Karlsson**[1]
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2025

Name of the student: Monika Witczak
Final project course: CSE3000 Research Project
Thesis committee: Jesse Krijthe, Rickard Karlsson, Ricardo Marroquim

Abstract

Estimating the Conditional Average Treatment Effect (CATE) with neural networks adapted for causal inference, like TARNet, is a promising approach, yet the impact of model architecture on performance remains underexplored. This paper systematically investigates how the depth and width of TARNet affect the CATE estimation in diverse simulated data environments. The research investigates two central questions: how TARNet's performance varies across data regimes (e.g., confounding strength, sample size), and how its optimal architecture changes in response to these conditions. A comprehensive set of simulation-based experiments is conducted using the CATENets framework, isolating and varying factors such as sample size, feature dimensionality, confounding strength, and the presence of noise. The results demonstrate that deeper architectures generally yield better performance in complex or high-dimensional scenarios, whereas narrower networks are preferable in small-sample or high-noise settings due to their regularizing effect. Furthermore, the findings suggest that there is no universally optimal architecture. The best configuration depends on the specific characteristics of the data. The study concludes with practical recommendations for architecture selection based on the experiments conducted.

1 Introduction

Causality is a relatively new notion in the field of machine learning; however, it is growing increasingly relevant. Estimating causal effects from observational data is a fundamental challenge in areas such as medicine, economics, and education, where understanding the effect of specific treatments or interventions (the so-called "treatment effect") is crucial for informed and effective decision making. In response, several state-of-the-art algorithms have been adapted to estimate causal inference. One such algorithm is neural networks, known for their powerful representation learning capabilities [1]. They have been successfully adapted to causal settings, offering a promising solution to estimating treatment effects through learned representations that mitigate confounding.

A key neural network model adapted for causality is the Treatment-Agnostic Representation Network (TARNet) introduced by Shalit et al. [2]. TARNet learns balanced representations to estimate treatment effects (especially Individual Treatment Effect, ITE) from observational data. However, in their work, the architecture of TARNet serves primarily as a tool to demonstrate theoretical generalization bounds, rather than being the central focus of empirical or architectural analysis.

In contrast, this research turns the attention toward a deeper empirical study of TARNet itself, particularly how architectural choices, such as the number of hidden layers and neurons per layer, influence the model's ability to estimate the Conditional Average Treatment Effect (CATE). Although CATE and ITE are conceptually related, CATE reflects average effects within subpopulations with specific conditions, rather than focusing on individuals [3], and thus is more generalizable for real-world scenarios.

Prior research in neural architecture optimization has demonstrated that model expressiveness - influenced by architectural hyperparameters like depth and width - significantly affects performance in standard ML tasks [4, 5]. However, little work has been done to understand how these design decisions affect performance in, specifically, causal inference settings, where challenges such as confounding, treatment imbalance, and limited overlap need to be taken into account in addition to overfitting or underfitting. Thus, while representation learning is central to modern causal ML models, the impact of how this representation is shaped by architecture remains underexplored.

That is why this research addresses the aforementioned knowledge gap by exploring the question:

How does varying the hyperparameters, specifically the number of layers and neurons per layer, in a TARNet neural network affect the performance of Conditional Average Treatment Effect (CATE) estimation on simulated datasets?

To answer this, a series of controlled experiments have been conducted using the CATENets framework [6, 7, 8], which includes a ready implementation of TARNet. By isolating architectural hyperparameters and analyzing their effect on CATE estimation accuracy using metrics such as Root Mean Squared Error (RMSE), this work aims to empirically characterize the influence of network design in a causal context.

The contribution of this work is thus mainly a systematic empirical study of how TARNet's architectural hyperparameters affect CATE estimation performance across simulated datasets of varying size and complexity. The results reveal that deeper architectures generally improve performance, particularly in high-dimensional or highly confounded settings, whereas narrower architectures offer better regularization and generalization in low-sample or high-noise conditions. No single architecture is universally optimal. Instead, performance depends on the interaction between model complexity and data characteristics. These findings culminate in practical recommendations for selecting architectures based on dataset properties, supporting future applications and research in neural causal inference.

The paper is organized as follows. The Background section introduces key concepts in causal inference and the TARNet model. The Methodology section outlines the experimental design and the data generation process. Experimental findings are presented in the Results section and interpreted in detail in the Discussion, which also addresses the study's limitations. Ethical considerations are discussed in the Responsible Research section. Finally, the paper concludes with a summary of findings and directions for future work in the Conclusions and Future Work section.

2 Background

Estimating causal effects from observational data introduces several challenges, including confounding, treatment imbalance, and the unobservability of counterfactual outcomes. To address these, machine learning methods such as TARNet have been proposed, which use neural networks to learn latent representations that support counterfactual prediction. This work focuses on estimating CATE, a target that offers a balance between personalization and statistical stability, particularly in noisy or high-dimensional settings.

This section provides the necessary context for the analysis that follows. It begins by motivating the use of CATE over ITE, outlines the main challenges of causal inference from observational data, and introduces the TARNet model within this setting. It concludes with a discussion of why architectural choices are critical for performance in neural causal inference and require further investigation.

2.1 CATE Estimation and Motivation

While ITE focuses on individual-level effects, this work studies CATE, defined as:

$$CATE(x) = \mathbb{E}[(Y_1 - Y_0) \mid X = x],$$

which represents the expected treatment effect for a subpopulation characterized by covariates X. This allows estimation of personalized treatment effects that remain generalizable between individuals with similar characteristics. As a result, CATE is often more stable and interpretable for real-world applications.

2.2 Challenges in Causal Inference from Observational Data

This study evaluates CATE estimators using simulated datasets, a choice driven by the Fundamental Problem of Causal Inference [9], which states that the true treatment effect $Y(1) - Y(0)$ for an individual is unobservable, as only one of the potential outcomes can ever be realized. Thus, evaluation using real-world data requires strong, often untestable assumptions. In contrast, simulated data provides direct access to both potential outcomes, allowing for reliable computation of estimation error using metrics such as RMSE of the CATE prediction, sometimes referred to as Precision in Estimating Heterogeneous Effects (PEHE) [10]. Although simulation-based evaluations can sometimes be overly optimistic by failing to capture certain complexities of real-world data (like treatment-confounder interactions or limited overlap) [6], they remain the standard for benchmarking causal estimators under controlled, transparent conditions.

Additional challenges arise due to the absence of randomized treatment assignment in observational data. The most relevant include:

Confounding: Confounders are variables that affect both treatment assignment and outcomes, introducing bias into effect estimates [11]. Since observational data lacks randomization, confounding is not naturally mitigated. Models like TARNet attempt to address this by learning latent representations, where treated and control groups become more balanced. However, this approach assumes that all relevant confounders are observed and included in the model - a condition known as "no unmeasured confounding."

Treatment Overlap: The positivity (or overlap) assumption requires that for every covariate profile X, the probability of receiving each treatment is strictly between 0 and 1 [12]. Violation of this assumption - for example, if treatment assignment is deterministic - can prevent reliable estimation of counterfactuals for certain subgroups. TARNet assumes that this condition holds for the learned representations to generalize effectively.

Treatment Prevalence: Imbalanced treatment assignment (e.g., 90% control and 10% treated) can skew model training, leading to overfitting on the majority group and underperformance on the minority group. Neural networks are especially prone to this issue unless regularization or architectural adaptations are applied.

2.3 TARNet Architecture and Causal Estimation

TARNet [2] is a neural architecture designed to estimate treatment effects from observational data under the potential outcomes framework (Rubin Causal Model [13]). The model is structured to learn a shared representation of covariates that supports accurate counterfactual prediction, even in the presence of treatment imbalance. It consists of the following components:

1. **Shared Feature Extractor:** A feed-forward neural network maps the input covariates X into a latent representation $\phi(X)$. This representation is shared across treatment groups and captures features relevant to both potential outcomes. The network is typically a multilayer perceptron (MLP) with a configurable number of layers and neurons per layer (which will be the main focus of this study), and uses non-linear activations such as ELU or ReLU.

2. **Treatment-Specific Outcome Heads:** Two separate MLPs, $h_0(\phi(X))$ and $h_1(\phi(X))$, take the shared representation as input and output the estimated potential outcomes under control ($\hat{Y}_0$) and treatment ($\hat{Y}_1$), respectively. For each data point, only the head corresponding to the observed treatment is used during training.

3. **Factual Loss:** The model is trained to minimize the prediction error on observed outcomes using a factual loss. For each individual i with covariates x_i, observed outcome y_i, and treatment assignment $t_i \in \{0, 1\}$, the prediction is made using the corresponding outcome head. To correct for imbalance in treatment group sizes, each sample is assigned a weight:

$$w_i = \frac{t_i}{2u} + \frac{1 - t_i}{2(1 - u)},$$

   where $u = \frac{1}{n}\sum_{i=1}^{n} t_i$ is the empirical proportion of treated units in the dataset. The factual loss is then computed as:

$$\mathcal{L}_{\text{factual}} = \frac{1}{n}\sum_{i=1}^{n} w_i \cdot \left(y_i - h_{t_i}(\phi(x_i))\right)^2.$$

   This weighting ensures that both treatment groups contribute equally to the learning objective, even if their prevalence is imbalanced.

4. **Regularization:** To prevent overfitting, the total loss includes an ℓ_2 penalty on the model parameters θ. The complete training objective is thus to minimize:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{factual}} + \lambda \cdot \|\theta\|_2^2,$$

   where $\lambda > 0$ controls the strength of the regularization.

Unlike its extension, CFRNet, TARNet does not include an explicit regularization term to minimize the distance between treated and control distributions in the latent space (e.g., via MMD or Wasserstein distance) [2]. Instead, it relies solely on shared representation learning and representation-balancing weighting. This makes TARNet architecturally simpler and easier to tune, as it avoids sensitivity to additional hyperparameters [6].

2.4 Importance of the Architecture

The architecture of the TARNet model, specifically, the number of layers and neurons, affects its ability to learn complex representations. Shallow networks may underfit, failing to capture nuanced patterns needed to mitigate confounding. Deep or wide networks, on the other hand, may overfit or become unstable, especially in low-data or noisy settings.

Although architectural choices are well studied in standard machine learning [4, 5], their role in causal inference remains underexplored. In particular, it is unclear how architectural complexity interacts with common challenges in causal settings, such as confounding, covariate imbalance, and limited sample size.

To address this gap, this work investigates the following questions:

1. How does TARNet's performance vary across different data regimes (e.g., confounding strength, input dimensionality, and dataset size) when using a fixed architecture?

2. How does the optimal TARNet architecture change in response to these data characteristics?

3. Based on the findings above, what practical recommendations can be made for selecting TARNet architectures under varying data conditions?

3 Methodology

Given the experimental nature of the research question, the methodology was structured accordingly to test the central hypotheses. The approach began with an initial literature review to establish a conceptual and theoretical foundation. Following this, a multi-stage experimental process was executed. The first stage involved reproducing established findings to validate the experimental framework. The subsequent stages included a comprehensive series of simulation-based experiments to answer the specific research questions. The methodology was concluded with a detailed analysis of the experimental outcomes, from which a set of general recommendations was derived.

3.1 Validation through Result Reproduction

A validation phase was conducted to ensure the reliability of the experimental framework utilized in this study. This involved reproducing selected results from key papers [2, 7], using the standard IHDP benchmark dataset [10]. The reproduction was performed using the CATENets framework [6, 7, 8], which is the same framework used for all subsequent experiments conducted in this research and can be found at `https://github.com/AliciaCurth/CATENets`.

The reproduction strictly followed the original experimental protocol, provided in [7], executing 100 realizations on the IHDP dataset via the authors' publicly available Python script. The resulting performance metrics were averaged and compared against the published values to confirm the accuracy of the implementation. Furthermore, a synthetic data experiment from the original research was replicated to verify that the data generating processes and the general trends in performance metrics, such as PEHE/RMSE, are aligned with expectations.

3.2 Experimental Design

A series of simulation-based experiments were designed to investigate the research questions. These questions concerned: (i) the performance of CATE estimations under various data regimes using a fixed neural architecture, and (ii) the identification of optimal neural architectures for specific, predefined data settings. All experiments utilized the synthetic data generating process, developed by Curth et al. for [6, 7, 8], detailed in Appendix B. Unless otherwise specified, default parameter values were used. Finally, to ensure robust results, all outcomes were evaluated using the RMSE of CATE estimation (PEHE), with performance metrics averaged across 10 independent simulation runs.

3.2.1 Performance Analysis with Fixed Architecture across Data Regimes

The initial experiments used a fixed, reference neural network and investigated how different data characteristics affect the CATE estimation performance. The main focus of this analysis was on the architecture utilized in [7, 2], which consists of three hidden layers for learning representations, each with 200 neurons. The remaining parameters have been set to default values, which can be found in Appendix A.

This model was then subjected to synthetic experiments, where the key parameters of the data generating process, such as sample size, the total number of covariates (dimensionality), treatment prevalence, as well as the number and strength of confounding variables, were systematically varied in isolation from each other. Here, the confounding strength refers to the strength of the relationship between covariates and the treatment assignment in the propensity score calculation, as described in Appendix B.2.

Additionally, two distinct experimental setups were designed to study the impact of confounding in further detail: one where the number of confounders increases within a fixed-dimensional space containing noise variables, and one where the feature space consists exclusively of confounding variables.

Furthermore, a T-Learner model was used as a baseline to provide a performance reference. Specifically, a TNet with the same network architecture as the primary model (3 layers, 200 neurons) was utilized. Contrary to TARNet, which learns a shared data representation, the T-Learner fits two entirely separate models to estimate the potential outcomes for the control and treated groups independently.

Finally, an experiment was designed to analyze how TARNet's performance changes when correlation is introduced among the covariates. To test this, simulations were run with an increasing dimensionality in two settings: default with uncorrelated (independent) covariates, and another with correlated covariates. Then the results were plotted together for a clear performance comparison.

3.2.2 Architectural Optimization for Fixed Data Settings

The second part of the research sought to determine the principles for selecting an optimal model architecture for a given dataset. These experiments explored how the interaction between model complexity and data

characteristics affects performance, particularly investigating the conditions under which more complex architectures might under- or outperform simpler ones. To this aim, a range of network architectures, systematically varying the number of hidden layers (from 1 to 5) and the number of neurons per layer (from 25 to 500, with values such as 25, 50, 100, 200, 300, and 500), were evaluated across several distinct experimental conditions.

One area of investigation concerned performance on small sample sizes. Experiments were run on training sets of limited size (e.g., 50 or 100 samples). It was hypothesized that more complex network architectures would be prone to overfitting on such limited data, resulting in higher out-of-sample error, compared to their simpler counterparts.

A second set of experiments examined the effect of low data dimensionality. Various low-dimensional datasets of 10 features were generated. These datasets represented different causal structures, such as containing only confounding variables (`n_c`), or a mix of confounding variables with either noise, heterogeneity variables (`n_t`), or prognostic variables (`n_o`). In this setting, highly complex network architectures were expected to overfit, despite a large sample size.

The impact of low confounding strength and minimal heterogeneity was also explored. An experimental setting was configured with no treatment effect heterogeneity (`n_t = 0`) and a very low confounding strength (e.g., $\xi = 0.1$), rendering treatment assignment nearly random. When applied to such low-signal data, the expectation was that complex models might overfit by attempting to model noise as meaningful patterns.

Finally, the effect of high dimensionality with varying causal structures was investigated. The experimental design included three feature space dimensions ($d = 30$, 50, and 100). The central hypothesis was that simpler, less expressive models would demonstrate greater robustness to a high volume of irrelevant features. More complex models were expected to be more susceptible to overfitting to this noise. To test this hypothesis, several data-generating processes were established. For a dimensionality of $d = 30$, three settings were analyzed: (i) a setting with default parameters ($n_c = 5$, $n_o = 5$, remaining $n_{\text{nuisance}} = 20$), (ii) a scenario composed entirely of confounding variables ($n_c = 30$), and (iii) a setting with 10 confounders (and 20 nuisance variables). For $d = 50$, the default parameter setting was evaluated, along with a higher confounding scenario with 25 confounders and 20 nuisance variables. Lastly, for $d = 100$, the default parameters were utilized, creating a challenging environment with a high volume of noise ($n_{\text{nuisance}} = 90$).

4 Results

The outcomes of the research are presented in this section. Following the initial validation through result reproduction, the core experimental findings are detailed, covering the performance of fixed TARNet architectures under diverse data conditions and the subsequent experiments on architectural optimization.

4.1 Reproduction of Previous Work

As outlined in the methodology, results from previous works [7, 2] were reproduced to verify the correctness of the implementation and to ensure the reliability of subsequent experiments based on the same framework. Focus was on Curth et al. [7] due to later utilization of their codebase.

On the IHDP dataset, reproduction yielded a PEHE of 0.698 (SE: 0.009) for the In-Sample setting and 0.704 (SE: 0.012) for the Hold-Out setting. These results are closely aligned with the reported values of 0.678 (SE: 0.009) and 0.689 (SE: 0.012), with only minor deviations likely due to stochastic effects or slight differences in experimental configuration.

Additionally, one synthetic experiment from [7] was reproduced to validate the simulation framework. As shown in Appendix C, the performance of SNet1 (TARNet) in experimental setting I matches the original results in both curve shape and numerical trends.

These outcomes confirm that the implementation is correct and can be reliably used in further experiments.

4.2 Performance Analysis with Fixed Architecture across Data Regimes

This section presents findings from experiments designed to assess a fixed TARNet architecture (SNet1: 3 layers, 200 neurons) under the various data regimes described in Section 3.2.1. The architecture was evaluated against a T-Learner (TNet) baseline through systematic alteration of parameters in the data generating process.

Many of the results are consistent with those presented in [7]; these confirmatory findings are summarized here (and detailed in Appendix D), as the main focus remains on novel simulations.

First, the impact of the number of predictive variables on the CATE estimation RMSE was considered (Figure 11 in Appendix D). A general trend was observed where a higher volume of predictive variables led to a higher RMSE. The prevalence of treatment was then examined, revealing a slightly U-shaped curve (Figure 13 in Appendix D). Optimal performance was observed for treatment proportions between 0.2 and 0.4, with RMSE increasing outside this range. Further experiments exploring sample size showed that a larger number of samples consistently reduced the RMSE (Figure 12 in Appendix D). In all these findings, which align with [7], TARNet consistently outperformed the simpler baseline in almost all cases.

The novel experiments focused on confounding strength and data dimensionality. As shown in Figure 1, increasing confounding strength from 0.0 to 10.0 led to a consistent rise in RMSE. A similar trend was observed for data dimensionality (Figure 2), where higher dimensions led to increased error. In both cases, the baseline model performed notably worse, with RMSE growing by approximately 1 on average.

The experiments also examined the effect of varying the number of confounding variables (n_c), initially while keeping the total number of features fixed at 25 (Figure 3). When no confounders were present ($n\_c = 0$), the model achieved low RMSE. Introducing a small number of confounders ($n\_c = 3$) led to a sharp increase in error, which gradually decreased as the number of confounders grew from 5 to 20. In contrast, the baseline model's performance improved only later, with a noticeable decline in RMSE at around 15 to 20 confounders.

To further explore the effect of confounding on estimation quality, an additional experiment was conducted in which the dimensionality consisted solely of confounding variables (Figure 4). The results showed a slightly U-shaped RMSE trend. Initially, the model achieved an RMSE of around 1, which reduced as confounders increased to about 10. The lowest RMSE occurred in the mid range (∼10 - 20 dimensions), after which the error started rising steadily. The baseline model, however, performed best with only three confounders. Beyond that point, the error increased sharply. By the highest dimensional setting, the performance gap between the two models was substantial - TNet exhibited an RMSE more than 2 points higher than SNet1.

Finally, an experiment assessed how covariate correlation impacts TARNet's performance when the dimensionality increases. The results (Figure 5) reveal two key findings: the model achieved a significantly lower RMSE when trained on correlated covariates across all dimensions, and the model's performance degraded at a much slower rate compared to the steep rise in error for the uncorrelated case.

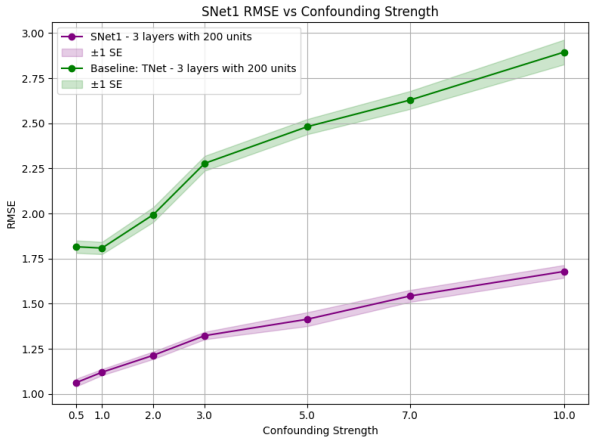


Figure 1: RMSE of the default (3 layers, 200 neurons) TARNet (SNet1) architecture against a simple baseline of TNet. The varied parameter is confounding strength at sample size n=2000. The shaded area represents one standard error.

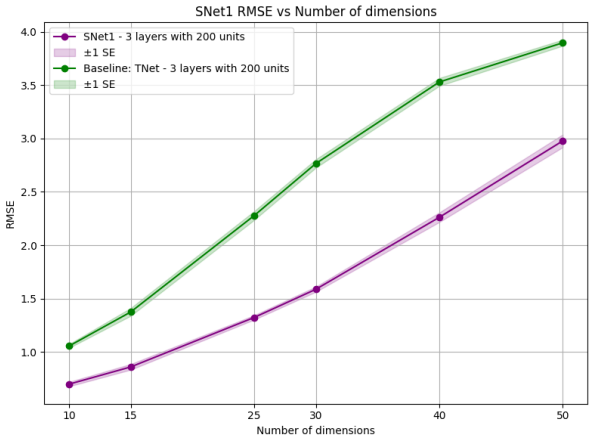


Figure 2: RMSE of the default (3 layers, 200 neurons) TARNet (SNet1) architecture against a simple baseline of TNet. The varied parameter is the dimensionality of data. The shaded area represents one standard error.

4.3 Architectural Optimization for Fixed Data Settings

A grid of 30 TARNet architectures was evaluated across multiple data regimes to explore the relationship between data characteristics and optimal model architecture. They varied in depth (1 to 5 hidden layers) and width (25 to 500 neurons per layer). The results are organized around specific data conditions and highlight how model complexity interacts with factors such as sample size, dimensionality, and confounding structure. While the summary of findings, outlining the most optimal architectures (Table 1) along with selected plots can be found in this section, full results are available in Appendix E.

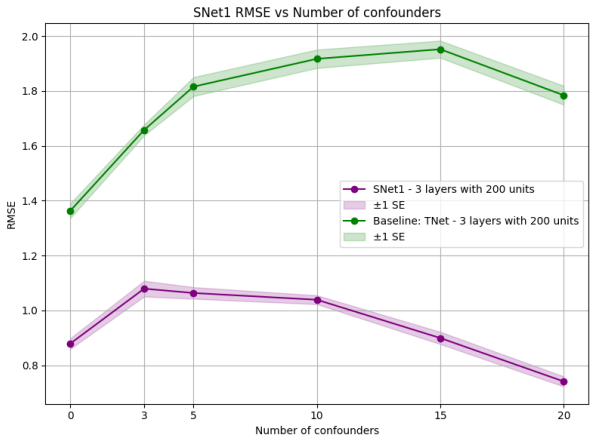


Figure 3: RMSE of the default TARNet (SNet1) architecture against a simple baseline of TNet, with varying number of confounders at a fixed dimensionality d=25. The shaded area represents one standard error.

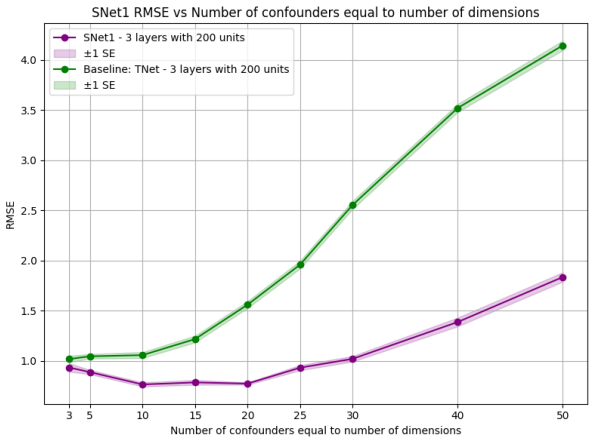


Figure 4: RMSE of the default TARNet (SNet1) architecture against a simple baseline of TNet, with a varying number of confounders equal to the dimensionality. The shaded area represents one standard error.

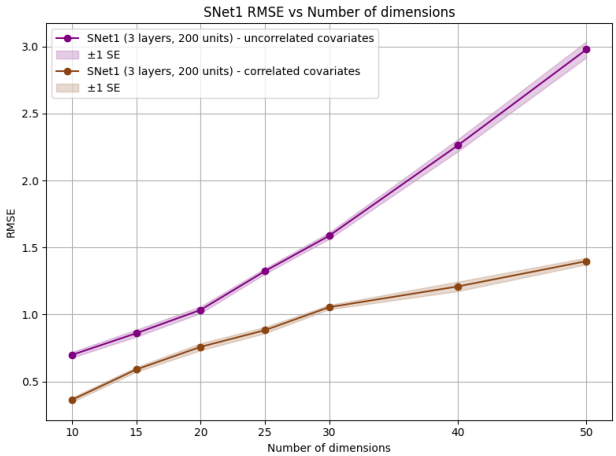


Figure 5: RMSE of the default TARNet (SNet1) across different numbers of dimensions in the data setting with correlated vs uncorrelated covariates. The shaded area represents one standard error.

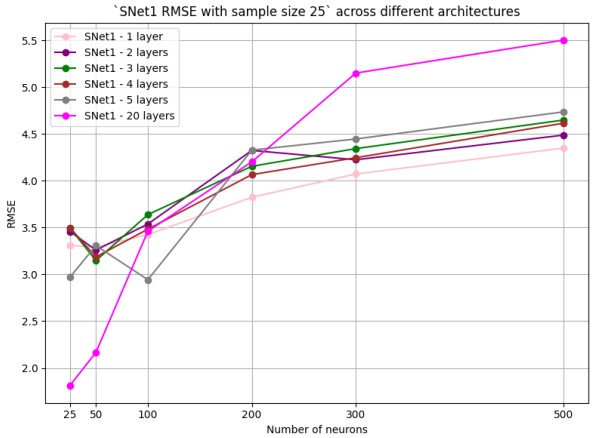


Figure 6: RMSE plot of different TARNet (SNet1) architectures for small sample size (here $n = 25$).

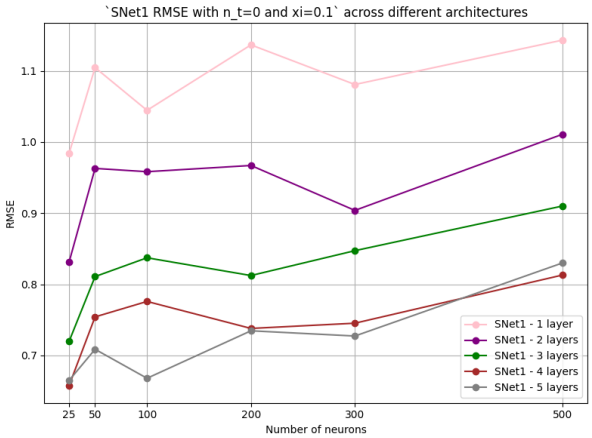
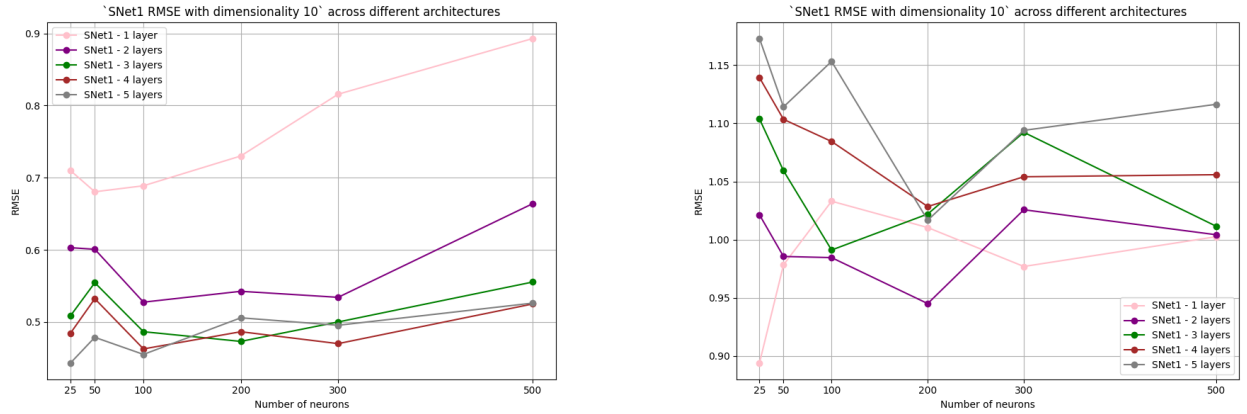


Figure 7: RMSE plot of different TARNet (SNet1) architecture in a low confounding strength scenario ($\xi = 0.1$).

Small Sample Sizes. When training data was limited (e.g., 25-500 samples), the performance was especially dependent on network complexity. Across all sample sizes, deeper networks (4-5 layers) consistently outperformed shallower ones (Figure 14 in Appendix E), but only when kept narrow. After these findings, an additional, considerably deeper, 20-layer network was introduced to observe if the same trend would still hold. In the extreme case of 25 samples (see Figure 6), the best-performing architecture was the 20-layer network with only 25 neurons. However, this architecture became unstable as the width increased. In contrast, 4-5 layer networks with 25 neurons showed more stable performance across widths, making them a safer choice in low-data settings. As the sample size increased to 500, wider networks became more viable, although the optimal configuration remained moderately deep and narrow.

Low Confounding Strength. In regimes with weak confounding ($\xi = 0.1$ to 0.5), deeper models (4-5 layers) again showed the best performance (as seen for example in Figure 7), particularly when combined with a narrow to moderately narrow width (25-100 neurons). Shallow architectures, on the other hand, consistently underperformed across all three values.



(a) Pure confounding scenario with 10 confounders ($d = n_c = 10$).

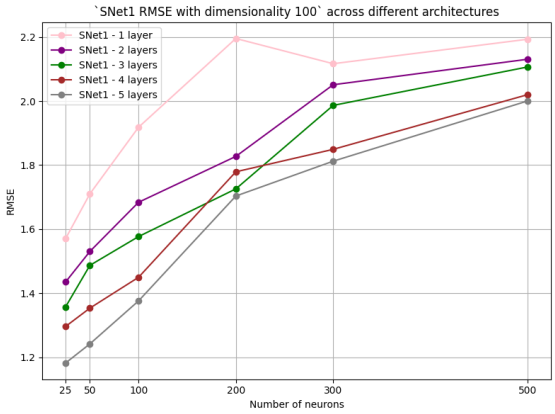
(b) Scenario with 5 confounders and 5 heterogeneity factors.

Figure 8: RMSE plots of different TARNet (SNet1) architectures for dimensionality $d = 10$.

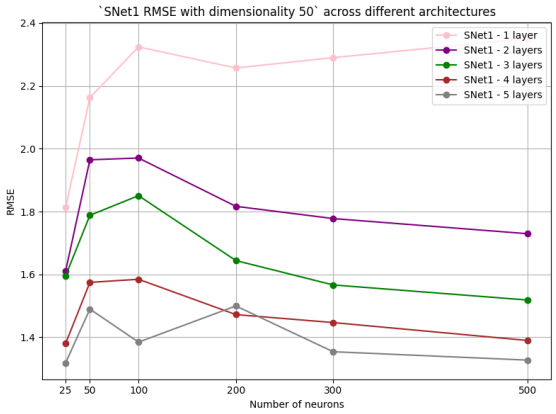
Low-Dimensional Data. Three different causal structures were explored in the scenario with 10 input features: equal (default) numbers of confounders and outcome predictors, only confounders, and a mix of confounders with noise variables. Across all three, deeper networks (3-5 layers) consistently achieved lower RMSE than shallower ones. Optimal performance was generally observed with 100 neurons, although narrower architectures (25 neurons) remained competitive. The best results came from a 5-layer network with 100 neurons. Notably, even in purely confounded settings (no outcome predictors), deeper architectures proved superior, as long as they were not overly wide (see Figure 8a).

Treatment Effect Heterogeneity. When treatment effect heterogeneity was introduced (i.e., with variables affecting the magnitude of the treatment effect) to the low-dimensional data setting, deeper networks no longer performed best, as seen in Figure 8b. The 1-layer network with 25 neurons and the 2-layer network with 200 neurons achieved the lowest RMSE. Deeper networks (e.g., 5-layer) achieved much higher RMSE across all neuron counts, showing signs of overfitting or difficulty generalizing.

High-Dimensional Data. High-dimensional scenarios (30, 50, or 100 features) presented challenges of noise and complexity. In cases where most features were irrelevant noise (e.g., 80-90%), only deep and narrow architectures performed well. For instance, in the 100-dimensional setting (Figure 9a), deeper networks (4-5 layers) with just 25 neurons achieved the lowest RMSE. Conversely, deeper and wider architectures became beneficial in settings with high signal (e.g., 25 confounders and 5 outcome predictors in 50 dimensions, as seen in Figure 9b). Here, the best performance was achieved by a 5-layer network with 500 neurons. This was even more noticeable in the extreme setting of 30 dimensions consisting solely of confounders, where most architectures drastically improved their performance with increasing the number of neurons.



(a) RMSE plot for $d = 100$ with high noise (90 noise variables), 5 confounders, and 5 outcome predictors.



(b) RMSE plot for $d = 50$ with 25 confounders, 5 outcome predictors, and 20 noise variables.

Figure 9: RMSE plots of different TARNet (SNet1) architectures for high dimensionality.

Table 1: Summary of best-performing TARNet architectures across different data regimes.

Setting	**Layers**	**Neurons**	**Observation**
Small sample size ($n = 25, 50, 100$)	4-5	25	Deep and narrow networks performed best. Shallow models underfit, wide ones overfit.
Moderate sample size ($n = 500$)	3-4	100-200	Wider networks became viable; moderately deep and wide architectures performed well.
Low dimensionality ($d = 10$)	5	100	Deep networks performed well across all variants (confounding only, noisy, outcome-relevant).
Treatment heterogeneity in low dimensionality ($d = 10$)	1-2	25, 200	Moderate depth and width offered best generalization; deeper networks began overfitting.
Low confounding strength ($\xi = 0.1, 0.3, 0.5$)	4-5	25-100	Deep networks remained effective; narrower widths reduced error in low-signal settings.
High dimensionality, low signal ($d = 100$)	5	25	Deep and narrow networks resisted overfitting to irrelevant features.
High dimensionality, high signal ($d = 50$)	5	500	Complex signal best captured by both higher depth and width.

5 Discussion

The experimental results reveal distinct patterns in how TARNet's performance for CATE estimation is influenced by various data characteristics while using different architecture settings. This section reflects on these findings and discusses them in the broader context of causal inference with neural networks. It first synthesizes the insights from evaluating fixed TARNet architectures across varying data regimes, then interprets the architectural optimization experiments. Finally, it devises a set of practical recommendations for model design and discusses key limitations of the study.

5.1 Performance with Fixed Architectures across Data Regimes

The results from evaluating a fixed TARNet architecture (3 layers, 200 neurons) across various synthetic data regimes confirm that data characteristics significantly influence model performance in CATE estimation. The observed rise in RMSE with increasing numbers of predictive variables aligns with expectations from standard ML. Not all predictive features help with causal estimation - many add noise, particularly when unrelated to treatment assignment or outcome generation. As such, the model's performance degrades with the inclusion of more non-causal predictors.

Treatment prevalence showed a slightly U-shaped relationship with RMSE. This can be attributed to the model's need for balanced exposure to both treated and control instances. When treatment prevalence is too low or too high, one group dominates the learning process, increasing estimation variance or bias.

Optimal performance was achieved when treatment prevalence ranged between 0.2 and 0.4 - an observation consistent with prior work on treatment overlap.

As expected, the analysis of sample size showed that larger training datasets consistently lowered the RMSE, highlighting the importance of sufficient data for accurate and stable CATE estimates.

More nuanced dynamics emerged in experiments that varied confounding strength and data dimensionality. As confounding strength increased, RMSE rose consistently, showcasing the increased difficulty of estimating treatment effects in the presence of stronger confounding. Similarly, increasing data dimensionality led to higher error, likely due to the model's difficulty in disentangling signal from irrelevant features in higher-dimensional spaces.

When varying the number of confounders and keeping total dimensionality fixed, the model exhibited a sharp increase in RMSE when a small number of confounders was introduced (e.g., $n_c = 3$), followed by a gradual decline as more confounders were added. This seems counterintuitive at first. However, this trend is likely due to the data generation process. As the proportion of confounders increases, the model receives more relevant information for estimating treatment assignment, while the amount of harmful noise decreases.

A similar but inverted trend appeared when dimensionality consisted entirely of confounders. The U-shape of the curve indicates that the model might be overfitting the initial very-low-dimensional setting. Later, in the mid-range, it finds the optimal range for its complexity. At some point, however, the performance starts degrading again due to underfitting and the curse of dimensionality.

A clear conclusion emerges from the comparison with the T-Learner baseline. TARNet consistently outperformed TNet in all settings. This is due to TARNet's shared representation of data for control and treatment groups, which appears to result in a more robust performance, compared to modeling each group separately, especially with increased data complexity (e.g., higher dimensionality and number of confounders).

Finally, the experiment concerning covariate correlation revealed that the performance of TARNet was consistently better and more robust to increasing dimensionality when the covariates were correlated (Figure 5). This can be explained by how TARNet's shared representation layer learns from the data structure. With uncorrelated features, each dimension provides new, independent information, making the learning task harder as the dimensionality grows. However, with correlated features, the network can exploit the redundancy that the correlation introduces. The shared layer does not need to learn separate signals for all dimensions. Instead, it can learn a more compact, lower-dimensional representation of the underlying factors that are relevant to the data. This makes the learning task easier and provides a strong regularization effect, reducing the curse of dimensionality and leading to better generalization.

5.2 Architectural Optimization Across Data Settings

The second phase of experimentation systematically varied TARNet's architectural parameters - depth and width - to identify optimal designs under different data regimes. Several patterns emerged.

First, depth consistently contributed to performance across most settings. Deeper networks (4-5 layers) generally outperformed shallower ones, especially in high-dimensional, noisy, or weakly confounded data. This finding is aligned with recent results in the deep learning literature, which suggest that deeper networks can model complex nonlinear structures more effectively than shallow ones [14, 15, 16].

Width, on the other hand, interacted more subtly with data properties. In low-sample settings (e.g., $n = 25$), wide networks exhibited severe overfitting, whereas narrow architectures (25-100 neurons) remained stable. Even extremely deep networks (20 layers) performed well if kept narrow. As the sample size increased, wider networks became viable, but only in combination with moderate depth (3-4 layers). This supports the view that width contributes capacity, while depth contributes representational power, and that both must be tuned to the complexity of data.

In low-dimensional data, deeper models still performed best, unless treatment effect heterogeneity was present. In such cases, shallower models (1-2 layers) with moderate width (100-200 neurons) provided better generalization. This suggests that modeling heterogeneity may require more caution - here, the performance cannot be improved by just increasing the model's depth. The network parameters should be chosen more carefully and focus on capacity.

In high-dimensional, low-signal settings (e.g., $d = 100$ with many nuisance features), only deep and narrow networks performed well, likely due to their regularizing effect, as wider models tended to overfit to noise. In contrast, in high-dimensional, high-signal settings (e.g., with many confounders or outcome predictors), both depth and width were beneficial, supporting the idea that increased model capacity is needed to capture more complex patterns. In these cases, the number of nuisance variables was more limited, allowing the model to benefit from additional width without overfitting.

An interesting point can be made about the prediction stability as well. In several settings, the architectures with the absolute lowest RMSE also seemed the most sensitive and potentially unreliable. For example, although a 20-layer network achieved the best performance with only 25 training samples, it was highly unstable across different widths (Figure 6). In contrast, moderately deep networks with 4 or 5 layers showed more consistent performance across neuron counts, making them a safer choice in low-data settings. A similar pattern emerged elsewhere, where a moderately wide network produced near-optimal results but with greater stability than the narrower architecture that achieved the best score. These findings point to a practical trade-off between optimizing for minimum error and selecting architectures that are more robust and dependable across a range of conditions.

5.3 Final Recommendations

Synthesizing both research questions leads to several practical recommendations for selecting TARNet architectures in causal inference tasks. These guidelines can support practitioners in choosing architectures more effectively, complementing established model selection techniques like cross-validation for hyperparameter tuning [17]. They also serve as a foundation for further research into optimal architectural design and model optimization in causal inference settings. The final recommendations comprise the following:

- **Generally use deeper architectures, but consider the context.** Start with deeper networks (e.g., 4-5 layers), as they generally perform better, especially with high-dimensional or noisy data. However, in low-dimensional settings with significant treatment effect heterogeneity, shallower models (1-2 layers) may offer better generalization.
- **Adjust network width based on sample size and signal quality.** For small datasets or data with many irrelevant features (high noise), use narrow networks (e.g., 25-100 neurons) to prevent overfitting. As the sample size or the proportion of relevant features increases, wider networks become beneficial for capturing more complex patterns.
- **Explore the role of correlated feature structures for TARNet.** Empirical results suggest that, contrary to conventional machine learning wisdom, correlated covariates can help TARNet learn more compact and generalizable representations, especially in high-dimensional settings. This potential property of TARNet warrants further investigation; nevertheless, exploring strategies for introducing or preserving relevant correlations through feature engineering (e.g., constructing interaction terms, informed feature selection) may still prove beneficial during model selection.
- **Consider stability, not just absolute best performance.** An architecture that achieves the lowest error might be unstable. Often, a moderately deep and wide model will provide near-optimal results with greater robustness across different data splits, making it a more reliable choice for practical applications.
- **Avoid unnecessary complexity.** In very low-dimensional or low-complexity scenarios, simpler architectures can outperform the overly complex ones. In these cases, the number of neurons might be a more impactful tuning parameter than depth.
- **Validate empirically on your specific data.** The key takeaway is that no single architecture excels in all situations. Always benchmark a few candidate models on your specific dataset. Do not blindly rely on default settings or findings from other studies, as performance is highly dependent on the unique characteristics of your data.

5.4 Limitations

Several limitations of this work should be acknowledged. Firstly, the experiments assume that all confounders are observed and correctly measured. This is a strong assumption, often violated in real-world applications. In the presence of unmeasured confounding, even the most well-tuned model may yield biased estimates.

Secondly, all results are based on synthetic simulations. While these provide valuable insights and ground truth for evaluation, they may not capture the full complexity or noise patterns present in real observational datasets. Furthermore, all the simulated data have been generated with particular data-generating processes, which assume specific function-based structures for the underlying data relationships. Consequently, if different functional forms were used to model these relationships, the qualitative nature of the results could change. Therefore, real-world benchmarking and exploration of different data generating processes would be important next steps.

Thirdly, this work focuses exclusively on TARNet as implemented in CATENets (SNet1). While it is a representative and widely studied model, results may not generalize directly to other architectures or causal frameworks. Moreover, hyperparameters beyond architecture, such as learning rate, regularization strength, and optimization schedule, were held constant to isolate architectural effects, but may also impact performance in practice.

Finally, a grid search was employed for architectural tuning, evaluating approximately 30 distinct TARNet configurations. While this is a substantial set, the possibility remains that more optimal architectures, not included in this defined search space, could exist.

6 Responsible Research

This research was conducted with a commitment to transparency, reproducibility, and ethical considerations. The entire experimental process was designed to be replicable, ensuring the validity of the findings. By detailing the methodology, including the use of the publicly available CATENets framework and providing specifics of the data-generating processes and parameters used (see Appendices A, B), this study adheres to the FAIR principles for data management [18] and upholds good research practices, as described in the Netherlands Code of Conduct for Research Integrity [19]. Following these rules enables other researchers to reproduce and verify the results, ensuring their reliability.

A central consideration in this work is the distinction between simulated and real-world data. This study relies on simulated datasets to allow for controlled and replicable experiments where the ground truth is known. However, it is crucial to acknowledge the future implications of this research, as the devised recommendations are intended for application in real-world scenarios, including high-stakes domains like medicine. Simulated data may not fully capture the complexities and biases of actual patient populations. When transitioning to real-world data, it is crucial to address potential biases, such as selection bias, which can arise if the study population does not represent the target population, leading to skewed and potentially harmful causal inferences. Therefore, sensitivity analysis, adjustment for selection bias, or other methods described in [20] should be undertaken to mitigate these negative effects. Moreover, further research is essential to validate the architectural guidelines on real-world benchmark datasets (e.g., IHDP, NEWS, JOBS, and TWINS) to analyze their practical efficacy.

Furthermore, as noted by Curth et al. in [6], current benchmarking practices for causal machine learning often utilize simulation settings that may be overly simplistic or not representative of real-world complexities. These simulations may fail to reflect significant challenges such as covariate shift, treatment assignment bias, or limited overlap. Consequently, while simulated benchmarks are necessary for evaluating estimator performance under controlled conditions, their results must be interpreted with caution and ideally complemented by analyses of robustness and realism.

This study aims to provide a solid foundation for future research in neural causal inference by systematically exploring the architectural choices within TARNet. The recommendations derived are intended to guide researchers in making more informed decisions, with the ultimate goal of improving the reliability and effectiveness of causal effect estimation in real-world applications. However, it is important to acknowledge the potential for these findings to be misused, a risk that can never be fully eliminated.

7 Conclusions and Future Work

This research systematically investigated the impact of network architecture on the performance of TARNet for Conditional Average Treatment Effect estimation. The study addressed two primary questions: how TARNet's performance with a fixed architecture varies across different data regimes, and how the optimal architecture changes in response to specific data characteristics like sample size, dimensionality, and confounding strength.

The findings conclude that there is no single universally optimal architecture for TARNet. It is instead highly dependent on the relations between model complexity and the specific characteristics of the data. Experiments revealed that deeper architectures (4-5 layers) generally outperform shallower ones, particularly in high-dimensional, noisy, or weakly confounded settings. However, the optimal width of the network is dependent on sample size and signal quality. For instance, in low-sample or high-noise scenarios, narrow networks (25-100 neurons) are crucial to prevent overfitting. Conversely, wider architectures become more beneficial as the sample size and the proportion of relevant features increase and the noise decreases, allowing the model to capture more complex patterns.

An interesting and recurring observation was the trade-off between achieving the absolute lowest error and ensuring model stability. In several experiments, the best-performing architecture was also highly

sensitive to slight neuron-count changes, and thus potentially unstable, whereas a moderately complex model offered near-optimal results with greater robustness.

Based on these results, this study offers practical recommendations for choosing optimal architectures for specific data tasks. The primary advice is to generally favor deeper networks, while adjusting width based on sample size and data quality - narrow for small or noisy datasets and wider for large, high-signal datasets. However, in cases of significant treatment effect heterogeneity, shallower models may provide better generalization. Ultimately, the findings show the necessity of empirical validation on the specific dataset in question, as performance is highly context-dependent.

For future work, several areas should still be explored. A crucial next step is to validate these architectural guidelines on real-world benchmark datasets such as IHDP, NEWS, JOBS, and TWINS to confirm if the same principles hold beyond simulated environments. Another direction would be a comparative analysis of TARNet against other models like DragonNet ([21]) or CFRNet ([2]) to understand how slight changes to neural network structure affect the quality of predictions. Finally, while this study isolated architectural effects, future research could investigate the combined impact of tuning other hyperparameters, such as learning rate and regularization strength, alongside network architecture.

A Default SNet1/TARNet Parameters

The default hyperparameters for the `SNet1` model, the implementation of TARNet in the CATENets codebase, are outlined below.

A.1 Model Architecture

- `n_layers_r`: 3 - The number of hidden layers in the shared representation block.
- `n_units_r`: 200 - The number of hidden units in each layer of the shared representation.
- `n_layers_out`: 2 - The number of layers in each of the two outcome heads.
- `n_units_out`: 100 - The number of units in each layer of the outcome heads.
- `nonlin`: `'elu'` - The activation function used in the neural network layers.
- `binary_y`: `False` - Whether the outcome variable is binary.
- `same_init`: `False` - Whether to initialize the two outcome heads with the same random weights.

A.2 Regularization

- `penalty_l2`: `0.0001` - The coefficient for the L2 (ridge) penalty on the weights.
- `penalty_disc`: `0` - The coefficient for the discrepancy penalty (MMD regularization) - this is not used for TARNet, only the extenstion CFRNet.
- `reg_diff`: `False` - Whether to apply a penalty on the difference between the outcome heads.
- `penalty_diff`: `0.0001` - The coefficient for the penalty on the difference between outcome heads (only used if `reg_diff` is `True`).

A.3 Training Parameters

- `step_size`: `0.0001` - The learning rate for the Adam optimizer.
- `n_iter`: `10000` - The maximum number of training iterations.
- `batch_size`: `100` - The number of samples per batch.
- `n_iter_print`: `50` - The number of iterations after which to print updates on the training progress.
- `val_split_prop`: `0.3` - The proportion of the training data to use for the validation set.
- `early_stopping`: `True` - Whether to use early stopping to terminate training.
- `patience`: `10` - The number of iterations to wait for improvement before stopping.
- `n_iter_min`: `200` - The minimum number of iterations before early stopping can occur.
- `seed`: `42` - The random seed for reproducibility.

B Data Generating Process Details

The simulation process is orchestrated by the `simulate_treatment_setup` function in the CATENets codebase (`https://github.com/AliciaCurth/CATENets`). This function generates a complete dataset of $n_{\text{train}} + n_{\text{test}}$ data points, which is then split into training and testing sets. By default, the sample sizes are $n_{\text{train}} = 2000$ and $n_{\text{test}} = 500$. The standard process for generating the components of these data points (covariates, potential outcomes, etc.) with default values is described below.

B.1 Covariate Generation (X)

The covariates $X \in \mathbb{R}^d$ are generated from a d-dimensional multivariate normal distribution:

$$X \sim \mathcal{N}(\mathbf{0}, \Sigma) \tag{1}$$

The total dimension $d = 25$ is composed of several parts:

- X_w: $n_w = 0$ covariates influencing treatment assignment only (instrumental variables).
- X_c: $n_c = 5$ covariates influencing both treatment assignment and outcomes (confounders).
- X_o: $n_o = 5$ covariates influencing outcomes only (prognostic variables).
- X_t: $n_t = 0$ covariates influencing the treatment effect only.(heterogeneity variables)
- X_{nuisance}: Other covariates with no influence on the process. ($n_{nuisance} = d - n_w - n_c - n_o - n_t$)

Thus, the full covariate vector is $X = (X_w, X_c, X_o, X_t, X_{\text{nuisance}})$. The covariance matrix Σ is either the identity matrix I_d (for uncorrelated covariates) or a generated correlation matrix.

B.2 Propensity Score Generation ($p(x)$)

The propensity score (the probability of receiving treatment) is modeled using the covariates that influence treatment assignment, $X_p = (X_w, X_c)$. A summary variable $z(X_p)$ is first computed. By default, this is a non-linear function:

$$z(X_p) = \frac{1}{n_w + n_c} \sum_{j=1}^{n_w+n_c} (X_{p,j})^2 \tag{2}$$

This initial score is then passed through the logistic (sigmoid) function, $\sigma(z) = \frac{1}{1+e^{-z}}$, to compute an intermediate propensity score, $p_{inter}(x)$:

$$p_{inter}(x) = \sigma(\xi \cdot z(X_p) + \text{offset}) \tag{3}$$

Here, 'offset' is a constant term. The parameter ξ controls the strength of the relationship between covariates and treatment assignment, essentially influencing how strong the confounders are, and thus for simplicity is referred to as **confounding strength** in this paper. For most experiments, this is set to $\xi = 3$.

Finally, to control the overall proportion of the treated population (**treatment prevalence**), the propensity scores are re-scaled to match a desired average propensity, $target\_p$. The final propensity score $p(x)$ is calculated as:

$$p(x) = p_{inter}(x) \cdot \frac{\text{target\_p}}{\mathbb{E}[p_{inter}(x)]} \tag{4}$$

where $\mathbb{E}[p\_inter(x)]$ is the average of the intermediate propensity scores across all individuals. This step ensures that the expected percentage of the population receiving treatment matches $target\_p$.

B.3 Potential Outcome Generation ($\mu_0(x), \mu_1(x)$)

The potential outcomes are generated based on different sets of covariates.

Outcome under control ($\mu_0(x)$)

This is determined by the confounding and prognostic variables, $X_y = (X_c, X_o)$. The default function is:

$$\mu_0(x) = \sum_{j=1}^{n_c+n_o} (X_{y,j})^2 \tag{5}$$

Treatment Effect ($\tau(x)$)

The CATE is determined by the covariates X_t. The default function is:

$$\tau(x) = \sum_{j=1}^{n_t} (X_{t,j})^2 \tag{6}$$

Outcome under treatment ($\mu_1(x)$)

This is modeled as the baseline outcome plus the treatment effect:

$$\mu_1(x) = \mu_0(x) + \tau(x) \tag{7}$$

B.4 Treatment Assignment (W)

The binary treatment assignment for each unit i is drawn from a Bernoulli distribution using the unit's propensity score:

$$W_i \sim \text{Bernoulli}(p(x_i)) \tag{8}$$

B.5 Observed Outcome Generation (Y)

Finally, the observed outcome Y is generated based on the "switching equation" (also known as the potential outcomes framework), with the addition of Gaussian noise:

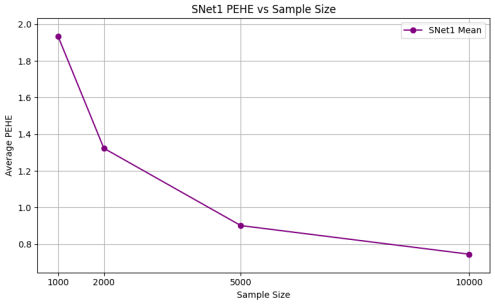
$$Y_i = W_i \cdot \mu_1(x_i) + (1 - W_i) \cdot \mu_0(x_i) + \varepsilon_i \tag{9}$$

where the noise term ε_i is drawn from a normal distribution with a mean of 0 and a standard deviation of σ_ϵ (referred to as `error_sd` in the code):

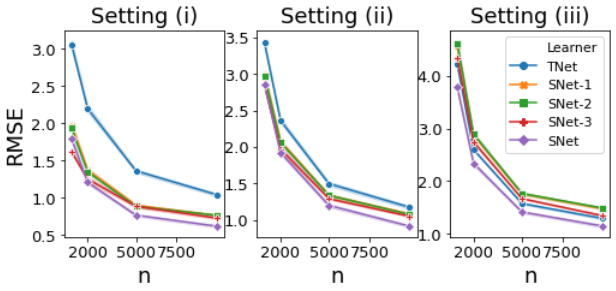
$$\varepsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2) \tag{10}$$

C Reproduction results

The following Figure 10 presents a comparison between the reproduced results and the original simulations. Specifically, Figure 10a shows the reproduced performance of SNet1 (TARNet) in setting I (as described in [7]), which is both shape- and value-wise comparable to the corresponding plot in Figure 10b, where the same model is depicted in orange.



(a) Reproduction of results from [7] in setting I using SNet1 (TARNet)



(b) Original results figure copied from [7] - settings I-III with TNet and all SNet models (SNet1 shown in orange)

Figure 10: Comparison of simulation results in setting I - average PEHE against the sample size shows the same curve and value trends in both original results and the reproduction

D Fixed Architecture across Data Regimes - Supplemental Results

This section presents supplemental findings that are aligned with results from prior work [7]. The experiments evaluate the performance of a fixed TARNet architecture (3 layers, 200 neurons) against a TNet baseline while systematically varying the number of predictive variables, sample size, and treatment prevalence.

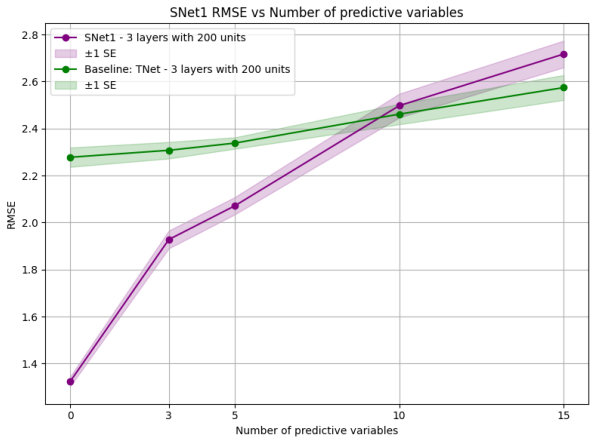


Figure 11: RMSE of the default (3 layers, 200 neurons) TARNet (SNet1) architecture against a simple baseline of TNet. The varied parameter is number of predictive features at sample size n=2000. Shaded area represents one standard error.

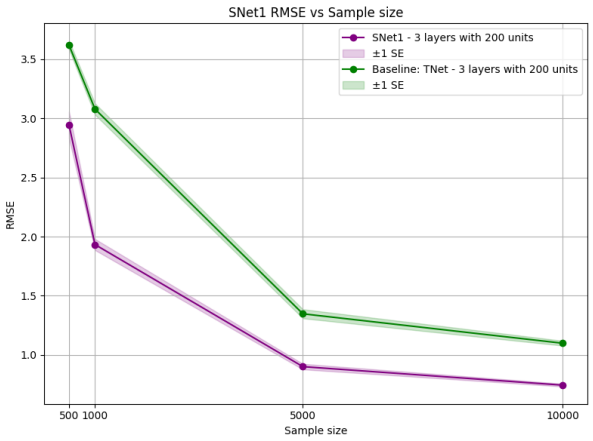


Figure 12: RMSE of the default (3 layers, 200 neurons) TARNet (SNet1) architecture against a simple baseline of TNet. The varied parameter is the sample size (for the training procedure). Shaded area represents one standard error.

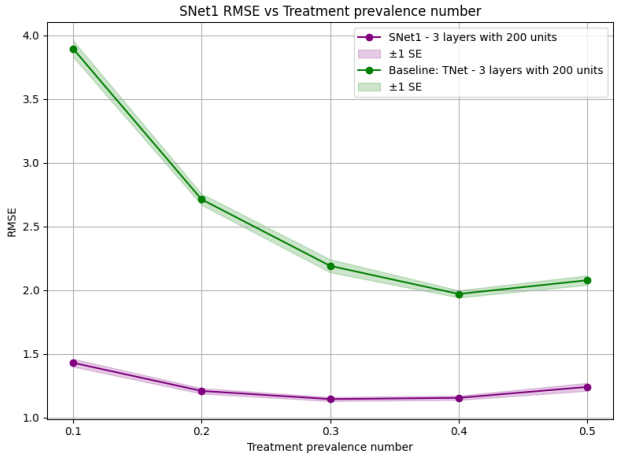


Figure 13: RMSE of the default (3 layers, 200 neurons) TARNet (SNet1) architecture against a simple baseline of TNet. The varied parameter is the treatment prevalence (proportion treated) at sample size n=2000. Shaded area represents one standard error.

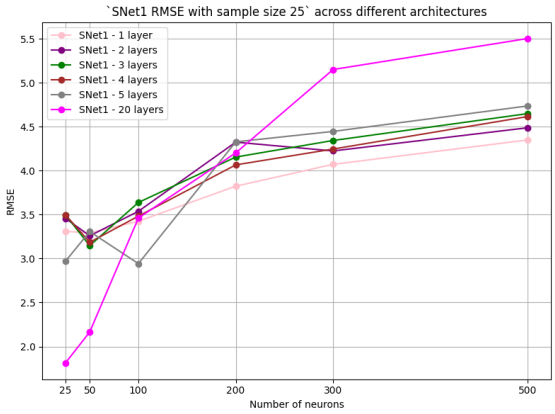
E Optimal Architecture across Fixed Data Regimes - Supplemental Results

This section provides the detailed results from the architectural optimization experiments discussed in the main paper. To investigate the relationship between data characteristics and the optimal model, a grid of TARNet architectures, varying in both depth (1 to 5 layers) and width (25 to 500 neurons), was evaluated across several fixed data regimes. The following subsections illustrate how model performance and the optimal architecture are influenced by factors such as sample size, data dimensionality, and confounding structure.

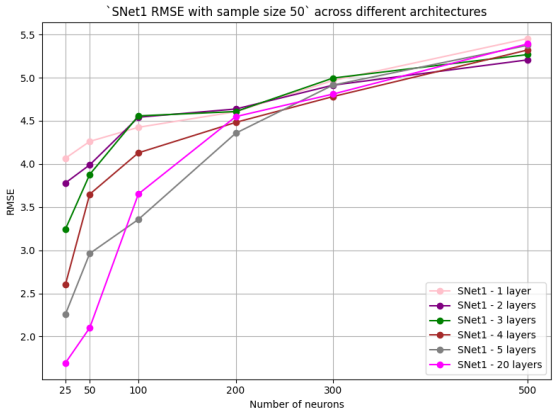
E.1 Influence of Sample Size on Model Performance

The hypothesis for this experimental subsection was that more complex architectures (i.e., those with more layers and neurons) might be prone to overfitting when the training dataset is small. Additionally, a drastically deeper neural network with 20 layers was included in the evaluation to observe its behavior with respect to overfitting.

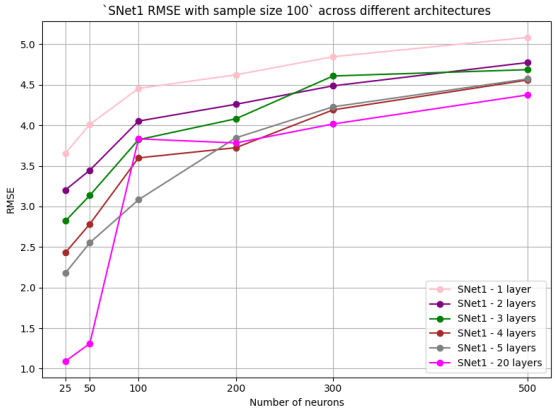
As illustrated in Figure 14a, a high RMSE was observed across all architectures. For nearly every layer configuration, the RMSE was found to increase with the number of neurons, indicating that wider networks



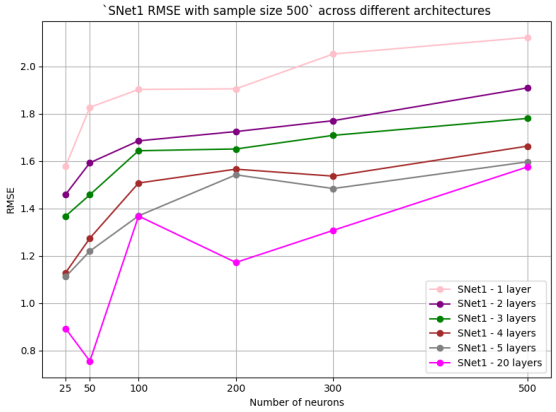
(a) PEHE plot for a small sample size of 25.



(b) PEHE plot for a small sample size of 50.



(c) PEHE plot for a small sample size of 100.



(d) PEHE plot for a sample size of 500.

Figure 14: PEHE plots for various small sample sizes.

quickly began to overfit. The additional 20-layer network exhibited the best performance, but only with a minimal number of neurons (25). It was concluded that with an extremely small sample size, the optimal approach is to employ a **deep but very narrow architecture**.

E.1.1 Sample Size of 50, 100, and 500

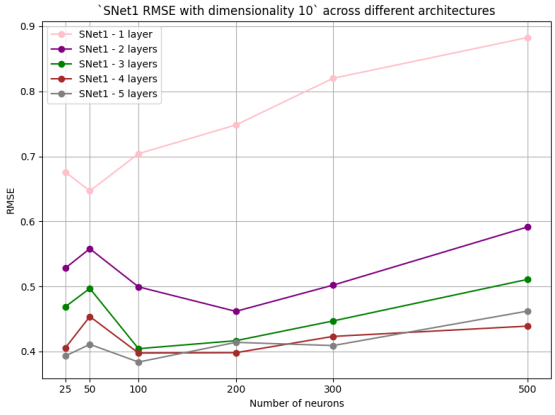
The trends for sample sizes of 50, 100, and 500 are shown in Figure 14b, Figure 14c, and Figure 14d, respectively. In all cases, performance generally worsened as the number of neurons increased, confirming the overfitting issue. With 50 samples, the 20-layer network with 25 neurons was again the top-performing model. As the sample size increased, the overall RMSE began to decrease, and the negative impact of adding more neurons became less severe, although the general pattern of deep and narrow architectures performing best was maintained.

E.2 Influence of Low Dimensionality

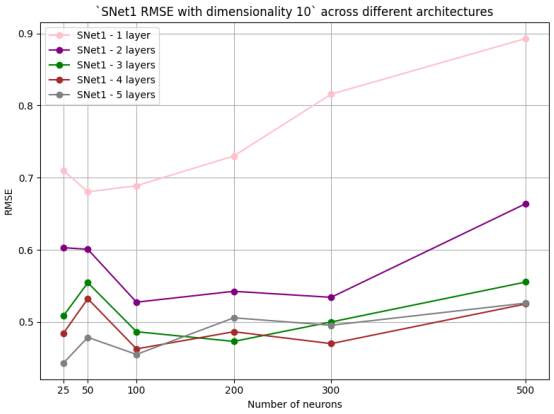
In this section, experiments were conducted using low-dimensional data ($d = 10$) with variations in the composition of the feature space. The other parameters were set to default values.

E.2.1 Dimensionality of 10: Mixed Feature Types

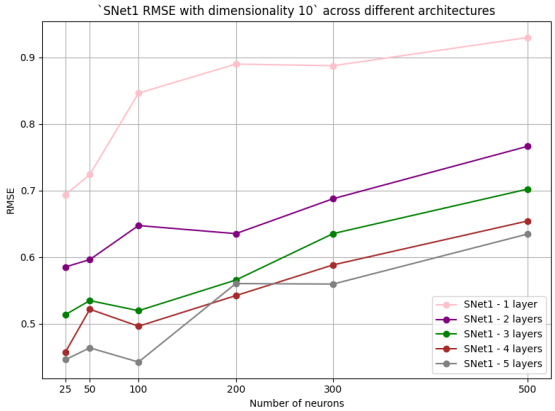
Several feature compositions were tested. With 5 confounders and 5 outcome predictors (Figure 15a), the **5-layer network with 100 neurons** achieved the lowest RMSE. In a "pure confounding" scenario with 10 confounders (Figure 15b), the optimal architecture was a **5-layer network with 25 neurons**. When the space included 5 confounders and 5 nuisance variables (Figure 15c), a **5-layer, 100-neuron** model was most effective. A noteworthy exception occurred with 5 confounders and 5 heterogeneity factors (Figure 15d), where a much simpler **1-layer model with 25 neurons** and **2-layer model with 100 neurons** achieved the lowest RMSE, indicating that excessive depth can be detrimental for certain tasks.



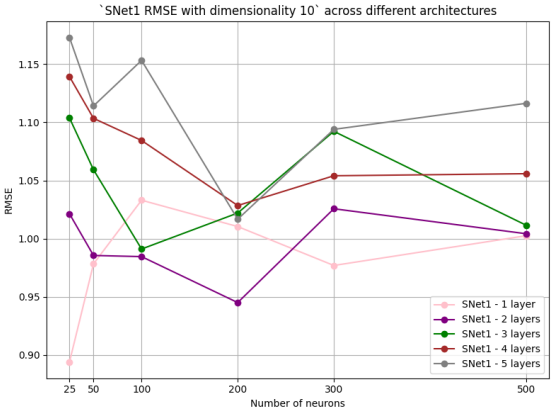
(a) PEHE plot for $d = 10$ with 5 confounders and 5 outcome predictors.



(b) PEHE plot for $d = 10$ with 10 confounders.



(c) PEHE plot for $d = 10$ with 5 confounders and 5 nuisance variables.

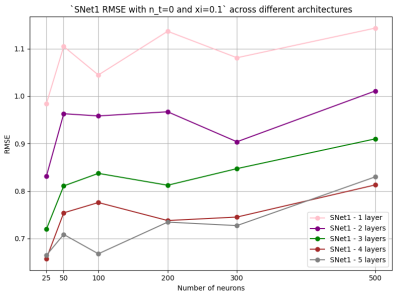


(d) PEHE plot for $d = 10$ with 5 confounders and 5 heterogeneity factors.

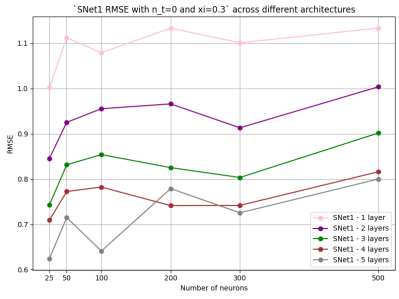
Figure 15: PEHE plots for $d = 10$ across different feature compositions.

E.3 Influence of Low Confounding and Signal

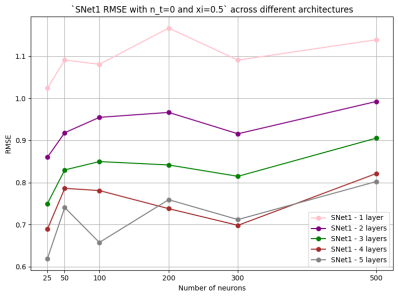
In this experiment, a low confounding strength (ξ) was used with no treatment effect moderators ($n_t = 0$). The results for confounding strengths $\xi = 0.1, 0.3$, and 0.5 are presented in Figures 16a, 16b, and 16c. Across all three experiments, a consistent pattern emerged: deeper architectures (4 and 5 layers) consistently achieved lower RMSE. With very low confounding ($\xi = 0.1$), the best performance was at a narrow width (25-100 neurons). As confounding strength increased, the optimal width also tended to increase, though narrow models remained highly competitive.



(a) PEHE plot for confounding strength $\xi = 0.1$.



(b) PEHE plot for confounding strength $\xi = 0.3$.



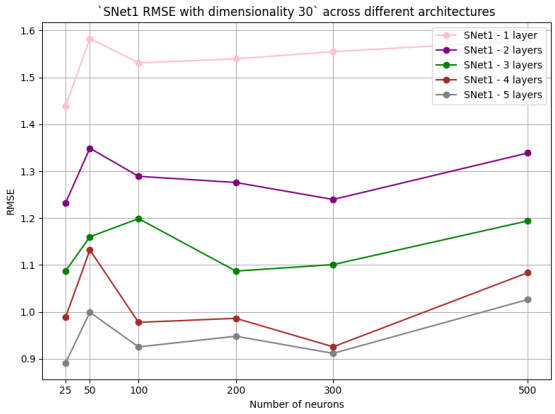
(c) PEHE plot for confounding strength $\xi = 0.5$.

Figure 16: PEHE plots for weak confounding strength across three ξ values.

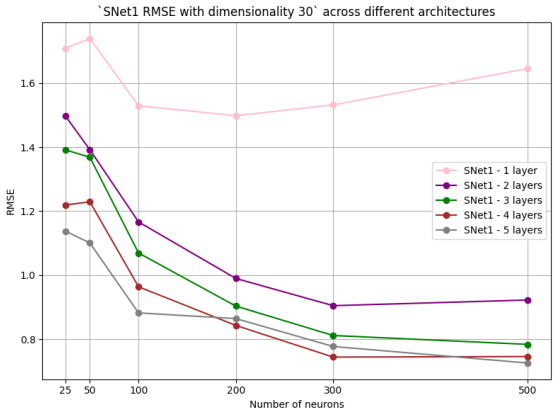
E.4 Influence of High Dimensionality

This set of experiments investigated model behavior in more complex, high-dimensional settings. When dimensionality was increased to 30 with a high proportion of noise variables (Figures 17a, 17c), deeper architectures remained superior, but with a moderate optimal width (25-300 neurons). However, when all 30 variables were confounders (pure signal, no noise, Figure 17b), performance improved with both deeper and **wider** networks, as the higher capacity was needed to model the complex signal, and there was no noise to overfit on.

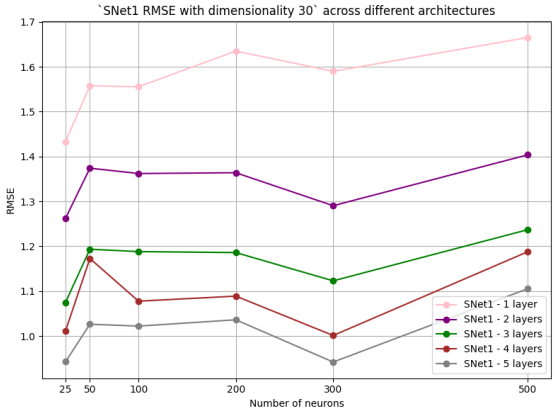
In extremely high-noise environments, such as a dimensionality of 50 or 100 with 80-90% noise variables (Figure 17d and Figure 17f), the optimal architecture was consistently deep but extremely narrow (25 neurons). Any increase in width led to immediate overfitting on the irrelevant features. Conversely, when the signal was strong (e.g., $d = 50$ with 25 confounders, Figure 17e), wider architectures again became beneficial.



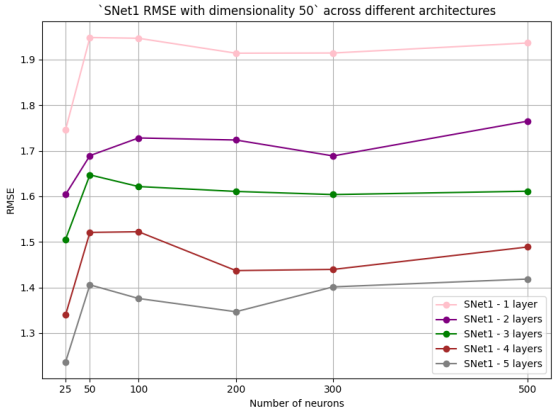
(a) PEHE plot for $d = 30$ with 5 confounders, 5 outcome predictors, and 20 nuisance variables.



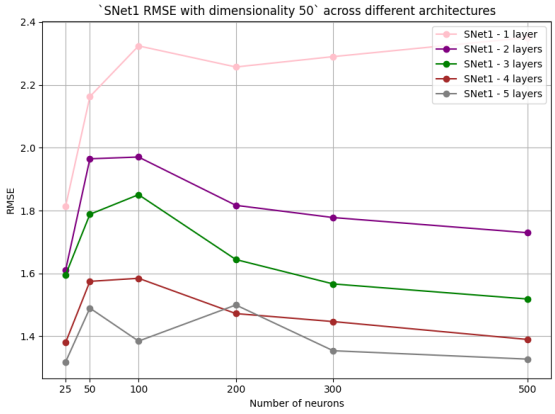
(b) PEHE plot for $d = 30$ with all variables as confounders.



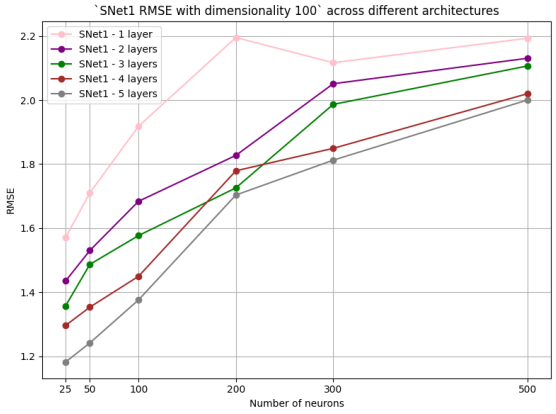
(c) PEHE plot for $d = 30$ with 10 confounders and 20 noise variables.



(d) PEHE plot for $d = 50$ with high noise (40 noise variables).



(e) PEHE plot for $d = 50$ with 25 confounders, 5 outcome predictors, and 20 noise variables.



(f) PEHE plot for $d = 100$ with very high noise (90 noise variables).

Figure 17: PEHE plots for various high-dimensional experimental setups.

F AI statement

Large Language Models (LLMs), including GPT-4, were utilized as assistive tools to refine the written expression of this paper. Their application primarily involved improving text for cohesiveness and flow (e.g., through prompts such as "Can you improve the cohesiveness and smooth over the following text: <text>?"). LLMs also assisted in lexical improvements, such as providing synonyms or rephrasing sentence components (e.g., "How can I replace the word 'XYZ' in the following sentence: <sentence>?"). Additionally, they provided help in formatting LaTeX code, especially improving the layout of Figures, e.g., with prompts

like: "How can I place two plots next to each other but make them separate Figures?". Crucially, the conceptualization, articulation, and intellectual content of all core ideas, arguments, and experimental results presented in this paper remain solely the work of the author.

References

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[2] Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International conference on machine learning*, pages 3076–3085. PMLR, 2017.

[3] Brian G. Vegetabile. On the distinction between "conditional average treatment effects" (cate) and "individual treatment effects" (ite) under ignorability assumptions. 2021.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[5] Renqian Luo, Fei Tian, Tao Qin, Enhong Chen, and Tie-Yan Liu. Neural architecture optimization. *Advances in neural information processing systems*, 31, 2018.

[6] Alicia Curth, David Svensson, James Weatherall, and Mihaela van der Schaar. Really doing great at estimating cate? a critical look at ml benchmarking practices in treatment effect estimation. 2021.

[7] Alicia Curth and Mihaela van der Schaar. Nonparametric estimation of heterogeneous treatment effects: From theory to learning algorithms. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2021.

[8] Alicia Curth and Mihaela van der Schaar. On inductive biases for heterogeneous treatment effect estimation. 2021.

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

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

[11] Lorenz Meuli and Florian Dick. Understanding confounding in observational studies. *European Journal of Vascular and Endovascular Surgery*, 55(5):737, May 2018.

[12] Stefan Feuerriegel, Dennis Frauen, Valentyn Melnychuk, Jonas Schweisthal, Konstantin Hess, Alicia Curth, Stefan Bauer, Niki Kilbertus, Isaac Kohane, and Mihaela Schaar. Causal machine learning for predicting treatment outcomes. *Nature medicine*, 30:958–968, 04 2024.

[13] Donald B Rubin and. Causal inference using potential outcomes. *Journal of the American Statistical Association*, 100(469):322–331, 2005.

[14] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016.

[15] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep networks with stochastic depth. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 646–661, Cham, 2016. Springer International Publishing.

[16] Monica Bianchini and Franco Scarselli. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *Neural Networks and Learning Systems, IEEE Transactions on*, 25:1553–1565, 08 2014.

[17] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[18] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J.G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A.C 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3(1):160018, Mar 2016.

[19] KNAW, NFU, NWO, TO2-federatie, Vereniging Hogescholen, and VSNU. Nederlandse gedragscode wetenschappelijke integriteit, 2018.

[20] C Keeble, GR Law, S Barber, and PD Baxter. Choosing a method to reduce selection bias: A tool for researchers. *Open Journal of Epidemiology*, 5(3):155 – 162, August 2015. © 2015 by authors and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY). http://creativecommons.org/licenses/by/4.0/.

[21] Claudia Shi, David Blei, and Victor Veitch. Adapting neural networks for the estimation of treatment effects. *Advances in neural information processing systems*, 32, 2019.