ResNCM: Uncovering Causal Drivers of STEM Academic Performance

Hanita Haller
Columbia University
hkh2122@columbia.edu

Aditya Nangia
Columbia University
an3325@columbia.edu

Eylam Tagor
Columbia University
et2842@columbia.edu

Abstract

Student performance in STEM disciplines is shaped by a variety of socio-economic, psychological, and behavioral factors, making causal inference a vital tool for disentangling true causal relationships from mere associations. In this paper, we introduce a novel residual neural architecture, ResNCM, which enhances Neural Causal Models by incorporating ResNet-style blocks for greater expressivity, stability, and support for categorical variables. We evaluate ResNCM on two rich educational datasets and demonstrate its superiority over classical machine learning and baseline deep learning models, achieving an accuracy of 89.98% on depression prediction and 93.39% on math score estimation. Beyond predictive performance, our model enables counterfactual reasoning to uncover actionable insights. Notably, we identify a direct causal effect of gender on math scores, where female students consistently outperform male peers, even after accounting for mediating and confounding variables. This work underscores the importance of integrating causal reasoning into educational AI and provides a scalable framework for future interventions. The presentation video can be found at the following google drive link

1 Introduction

Educational outcomes in Science, Technology, Engineering, and Mathematics (STEM) fields are influenced by a web of interdependent factors, ranging from psychological well-being and family background to socio-economic status and school-level variables. Traditional statistical and deep learning approaches often conflate correlation with causation, leading to misleading insights that may reinforce existing inequities rather than mitigate them. In this work, we seek to go beyond predictive modeling by leveraging causal inference to uncover actionable pathways that directly impact student performance in STEM domains.

The need for causal reasoning in educational data analysis is paramount. Consider the case where students from higher socio-economic backgrounds perform better not due to innate ability, but because of access to private tutoring or greater study time—resources not uniformly distributed. Observational models may misattribute performance gains to immutable traits like demographic features, rather than underlying structural causes such as academic support or mental health. Causal models, on the other hand, enable counterfactual reasoning: we can ask, for instance, how a student's grades might have changed had they received proper test preparation or faced less work pressure. This capacity is essential for fair and effective policy interventions.

Despite its promise, causal inference remains underutilized in practical machine learning applications due to its theoretical complexity and strict identifiability assumptions. Recent advancements in Neural Causal Models (NCMs) offer a bridge between the rigor of causal inference and the scalability of deep learning. Introduced in Xia et al. [2022], NCMs generalize Structural Causal Models (SCMs) by

Preprint. Under review.

representing each structural equation with a neural network, allowing for expressive and data-driven discovery of complex causal relationships, including counterfactuals.

In this paper, we implement a robust NCM framework on two real-world-inspired educational datasets to explore the true drivers of student success in STEM. Our key contributions include:

- A comprehensive preprocessing and feature engineering pipeline for two high-dimensional, noisy educational datasets.
- A novel ResNet-based mode, ResNCM, capable of estimating counterfactual outcomes and isolating direct, indirect, and spurious effects.
- Empirical evaluations comparing NCMs with classical and deep learning models, demonstrating superior accuracy and interpretability.
- Counterfactual case studies revealing how modifiable variables (e.g., study hours, test preparation) causally affect academic performance.

The remainder of the paper is organized as follows. In Section 2, we discuss the theoretical foundations of causal inference and review related work. Section 3 describes the datasets and preprocessing procedures. Section 4 outlines our model architecture, training strategies, and causal graph design. Section 5 presents our experimental results, including counterfactual analyses. Finally, Section 6 summarizes our findings and outlines future directions.

2 Background and related work

This paper's contribution is a unique and thorough implementation of the Neural Causal Model, which is at the intersection of Artificial Neural Networks and Judea Pearl's Causal Inference, modified to handle large amounts of data and infer complex causal relationships. This section highlights significant works that served as the theoretical backbone of our model.

Structural Causal Models and the Causal Hierarchy. Structural Causal Models (SCMs) are the primary unit of formal language to encapsulate an environment's causal mechanisms. An SCM $\mathcal{M}$ consists of:

- V: a list of the endogenous (observed) variables.
- U: a list of the exogenous (unobserved) variables.
- $\mathcal{F}$: a list of functions that define the dependencies of each endogenous variable's value. The input variables for f_v are all parents of v.
- $P(U)$: the probability distribution of the exogenous variables' values.

The SCM induces the Pearl Causal Hierarchy, which consists of three layers: observational (L1), interventional (L2), and counterfactual (L3) Pearl and Mackenzie [2018]. Although our work primarily invokes L3, knowledge of the other two layers is essential to conducting meaningful analysis of the datasets' causal relationships.

Fairness. In this paper, we use the Standard Fairness Model developed in Plecko and Bareinboim [2022], which provides a framework for determining direct effects, indirect effects, spurious effects, etc. in parameterized datasets. As detailed in Section 4, we designed an expanded fairness model that maintains the necessary causal relationships in high-dimensional environments.

Neural Causal Models for Counterfactual Inference. The Neural Causal Model (NCM) 1 introduced in [Xia et al., 2022] is a wrapper for the SCM that replaces each variable's function in $\mathcal{F}$ with a feed-forward neural network. The endogenous and exogenous variables V and U remain the same, and the exogenous distribution $P(U)$ is sampled from simple priors. In [Xia and Bareinboim, 2024], it is proven that any NCM with a specified causal diagram $\mathcal{G}$ automatically satisfies all L3 (counterfactual) equality constraints implied by $\mathcal{G}$. Although the architecture described in the paper is minimal and fit for theoretical "toy" experiments, it was the starting point of this paper's implementation of an NCM for counterfactual inference in real data that has complex causal mechanisms.

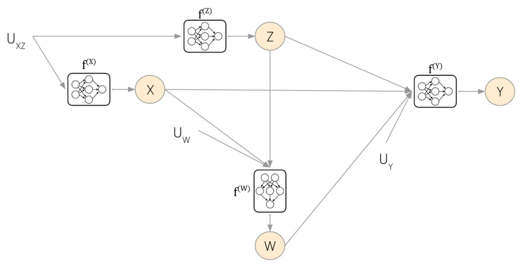

Figure 1: Neural Casual Model

Expressiveness vs. Learnability in Causal-Neural Models. The significance of the NCM for causal inference was demonstrated in [Xia et al., 2021]. More specifically, Xia et al. showed that universal approximation alone is insufficient for bridging the gap between observational fitting and either interventional or counterfactual reasoning by applying the Causal Hierarchy Theorem to a neural setting. Through these contributions, we were driven to apply the nominal concept of the NCM in a significant deep learning setting to draw meaningful conclusions about data that other deep learning approaches are now proven to be incapable of drawing.

Other Neural Approaches to Causal Inference. Beyond NCMs, various deep-learning approaches have been proposed for causal effect estimation under stronger assumptions. Representation-learning methods leverage balancing networks or domain-adversarial training to adjust for confounding under backdoor conditions [**??**]. Generative models—including GANs [**?**], normalizing flows, and variational autoencoders—have been employed to model interventional distributions in Markovian settings [**??**]. However, these methods typically focus on L2 estimation under no-unobserved-confounders or specific graph structures, and do not address general counterfactual identification in non-Markovian SCMs. On the other hand, this paper conducts meaningful L3 analysis even with a loose and flexible graph structure, and in the presence of unobserved confounders.

Our work builds directly on these two threads: we expand on the NCM framework of Xia et al. [2022] and the theoretical insights of Xia et al. [2021], extending them to STEM education datasets. In particular, we conduct counterfactual analyses of academic performance, demonstrating how NCMs can yield actionable insights in real-world, high-dimensional educational settings that non-causal machine learning approaches cannot.

3 Dataset Description

3.1 Dataset 1: Student Depression Dataset

To model causal relationships affecting student performance in STEM fields, we selected the *Student Depression Dataset* sourced from Kaggle B. [2022]. This dataset provides rich contextual information on psychological, academic, and demographic factors, making it particularly suitable for causal inference in educational settings.

The dataset comprises records of individual students, with each row representing a unique student and each column detailing a specific attribute. Key features include:

- **Demographics**: Age, Gender, City
- **Academic performance**: CGPA, Study Satisfaction, Work/Study Hours
- **Lifestyle**: Sleep Duration, Dietary Habits
- **Psychological indicators**: Depression Status (target variable), Suicidal Thoughts, Family History of Mental Illness
- **Stress and pressure**: Academic Pressure, Work Pressure, Financial Stress

The target variable is `Depression_Status`, a binary label indicating whether a student is experiencing depression (`Yes` or `No`), serving as a proxy for mental health status that may influence STEM performance outcomes.

Benefits of Dataset Selection

This dataset was chosen for several key reasons:

- **Multidimensional Factors**: It captures a wide array of variables influencing both mental health and academic performance—crucial for identifying confounding and mediating variables in causal inference.
- **Granularity and Diversity**: The inclusion of diverse educational backgrounds, lifestyle choices, and psychological factors enables detailed subgroup analysis within the STEM domain.
- **Applicability to Interventions**: Understanding causal pathways through this data can inform actionable policies for improving student well-being and academic outcomes.

Cleaning and Preprocessing

The raw dataset required substantial preprocessing to ensure consistency, interpretability, and suitability for downstream neural causal modeling tasks:

1. **Encoding Categorical Variables:**
   - Gender was mapped to binary values (`Male` = 0, `Female` = 1).
   - Sleep duration was binarized (`< 5 hours` = 0, otherwise = 1).
   - Dietary habits and family history of mental illness were encoded as binary values.
   - Suicidal ideation was transformed into a binary feature.
2. **Numerical Type Casting:** Features such as `Age`, `Study Satisfaction`, `Work/Study Hours`, and `Financial Stress` were cast to integer types to ensure compatibility with numerical models.
3. **Educational Background Representation:** Degree programs were first mapped to their full-text equivalents. Two categorical abstractions were derived:
   - *Degree Level* (e.g., undergraduate, postgraduate, doctoral)
   - *Domain* (e.g., Technology, Science, Arts)
4. **Text Embedding and Clustering:** Degree names were embedded using a pre-trained transformer model (`all-MiniLM-L6-v2`) from SentenceTransformers. The resulting 384-dimensional vectors were clustered using K-Means ($k = 4$), introducing a new feature, `degree_cluster`.
5. **Dimensionality Reduction:** To enable interpretable modeling, PCA was applied to the degree embeddings, and the top 5 principal components were retained as new

3.2 Dataset 2: Students Exam Scores Dataset

The second dataset used in our study is the *Students Exam Scores Dataset*, obtained from Kaggle Kimmons [2023]. This dataset includes standardized test scores in three subjects—Mathematics, Reading, and Writing—along with a variety of socio-economic and personal factors that potentially influence academic performance. The dataset is synthetic and intended for educational purposes, consisting of over 30,000 records, which significantly exceeds the size of similar publicly available datasets.

This extended version includes 15 distinct features, many of which exhibit missing values, making it ideal for testing preprocessing pipelines and robustness in neural causal inference models.

Each row in the dataset corresponds to a unique student, with features covering:

- **Demographics and Family Background**: Gender, Ethnic Group, Parental Education, Parent Marital Status, Is First Child, Number of Siblings
- **Socio-economic Status**: Type of Lunch, Weekly Study Hours, Transport Means
- **Behavioral Factors**: Test Preparation Completion, Sport Practice Frequency
- **Academic Performance**: Scores in Math, Reading, and Writing (0–100 scale)

These factors provide a holistic view of a student's learning environment and behavior, supporting the discovery of nuanced causal relationships that affect STEM outcomes.

Benefits of Dataset Selection

Key advantages of using this dataset include:

- **Large Sample Size**: With over 30,000 samples, the dataset allows for high statistical power in both causal discovery and effect estimation.
- **Diverse Attributes**: It includes categorical and ordinal variables covering psychological, social, and academic dimensions—critical for disentangling causal pathways.
- **Realistic Noise and Missingness**: The presence of incomplete records reflects real-world data conditions, making this dataset suitable for evaluating the robustness of preprocessing and inference methods.

Cleaning and Preprocessing

To prepare the data for causal analysis, the following preprocessing steps were applied:

1. **Column Removal:** An unnecessary index column (`Unnamed: 0`) was dropped from the dataset.
2. **Encoding Categorical Variables:**
    - `Gender`: Encoded as binary (`male` = 0, `female` = 1)
    - `EthnicGroup`: Mapped from group A–E to integers 0–4
    - `ParentEduc`: Categorized based on educational attainment:
        - High school or less = 0
        - Some college or associate's degree = 1
        - Bachelor's degree = 2
        - Master's degree = 3
    - `LunchType`, `TestPrep`, `IsFirstChild`, `TransportMeans`: All converted to binary indicators
    - `ParentMaritalStatus`: Encoded as ordinal categories:
        - Single = 0, Divorced = 1, Widowed = 2, Married = 3
    - `PracticeSport`: Mapped to frequency scale (Never = 0, Sometimes = 1, Regularly = 2)
    - `WklyStudyHours`: Discretized into ordinal scale:
        - $<$5 hrs = 0, 5–10 hrs = 1, $>$10 hrs = 2
3. **Missing Values:** Basic imputation strategies were considered for handling missing data (e.g., mode or median imputation), though more advanced imputation may be performed in later modeling stages.

This preprocessing schema ensures that the dataset is numerically encoded and cleansed, enabling its seamless integration with deep causal models for estimating the effects of socio-economic and behavioral factors on STEM performance outcomes.

4 Methodology

For each dataset, we start by creating a projection of the data features onto the standard fairness model and initializing our neural causal model based on the resulting graph. We then process the data, and train our NCM on the processed data. Once trained, we are able to evaluate counterfactual queries from the model.

4.1 Model Architecture

Our Neural Causal Models (NCMs) are built by instantiating one neural network per node in the causal graph $\mathcal{G}$, each network f_v learning the structural equation

$$v \;=\; f_v\big(\mathrm{Pa}_{\mathcal{G}}(v),\, U_v\big)\,,$$

where $\mathrm{Pa}_{\mathcal{G}}(v)$ are the parent variables in $\mathcal{G}$ and U_v is exogenous noise. We compare two instantiations of these per-node networks:

Feed-Forward Neural Causal Model (FF-NCM). Following Xia et al. [2022], each f_v is a vanilla multi-layer perceptron (MLP) implemented in `mlp.py` and wired up in `feedforward_ncm.py`. Concretely:

- Inputs: concatenation of parent embeddings $\{v_i\}$ and noise samples $\{U_{v_i}\}$.
- Architecture: a stack of h_layers linear layers (default 2), each of width h_size (default 128), with LayerNorm + ReLU activations, followed by an output layer and a Sigmoid.
- Initialization: Xavier normal on all linear weights.
- Noise prior: uniform over $\{U_v\}$.

This FF-NCM is simple and broadly applicable, but can suffer from limited depth making it harder to model very complex interactions, and from having no specialized support for high-cardinality categorical parents.

Residual Neural Causal Model (ResNCM). To address these limitations, we developed a residual-block variant (`resnet.py` and `residual_ncm.py`) in which each f_v is implemented as a ResNet:

- **Categorical embeddings**: for any discrete parent p, we learn an embedding Embed_p of size d_p.
- **Input projection**: numeric parents, embeddings, and noise are concatenated into $\mathbf{x} \in \mathbb{R}^{i\_size}$, then linearly projected to a hidden residual signal $\mathbf{x}_{\mathrm{proj}}$.
- **i-layer core with skip**:

$$\mathbf{h}_i = \mathrm{ReLU}\big(\mathrm{LN}(W_i\mathbf{x})\big), \quad \ldots, \quad \mathbf{r} = \mathrm{Dropout}\big(\mathbf{h}_n + \mathbf{x}_{\mathrm{proj}}\big).$$

- **Output head**: a final linear layer plus Sigmoid produces the predicted v.
- Hyperparameters: hidden size h_size (default 128), dropout (default 0.1).

The residual connection and LayerNorm ensure stable gradient flow even as depth increases, and the learned embeddings enable compact, expressive handling of categorical variables. This is especially important for this paper's objective of applying NCMs to complex, high-dimensional data.

Comparison and Advantages.

- **Expressivity:** ResNet blocks capture higher-order interactions via deeper representational paths, whereas the vanilla MLP is shallower.
- **Gradient stability:** skip connections mitigate vanishing/exploding gradients, speeding convergence in ResNCM.
- **Categorical support:** FF-NCM treats all inputs numerically; ResNCM explicitly embeds discrete parents, reducing parameter waste on one-hot inputs.
- **Regularization:** Dropout in ResNCM improves robustness to overfitting, especially in smaller datasets.
- **Plug-and-play:** Both models share the same SCM interface (`SCM` base class) and noise prior; swapping in ResNet layers requires only changing the 'f' module in `ResNCM` versus `FF_NCM`.

Empirically (see Section 5), our ResNCM consistently achieves faster training convergence, higher accuracy, and higher confidence on both outcome and feature predictions compared to the FF-NCM baseline.

Causal Diagram. Successful implementation of a Neural Causal Model requires not only a robust neural network architecture for learning the functions, but also a causal diagram $\mathcal{G}$ that defines the causal relationships between variables and ultimately induces the SCM. We created an expanded, flexible version of the Standard Fairness Model Plecko and Bareinboim [2022] that is capable of handling high-dimensional data without sacrificing the causal relationships, represented by directed edges for effects and bidirected edges for confounding.

4.2 SFM Projection

We started by projecting our real data features onto the SFM 2.

Depression Projection

X Gender

Z Age, Sleep Duration, Family History of Mental Illness

W Academic Pressure, Work Pressure, CGPA, Study Satisfaction, Dietary Habits, Suicidal Thoughts, Work/Study Hours, Financial Stress, Degree Level, Degree Cluster

Y Depression

Exam Scores Projection

X Gender

Z Ethnic Group, Parent Marital Status, Is First Child, No. Siblings, Transport Means

W Parent Edu., Lunch Type, Test Prep, Practice Sport, Weekly Study Hours, Reading Score, Writing Score

Y Math Score

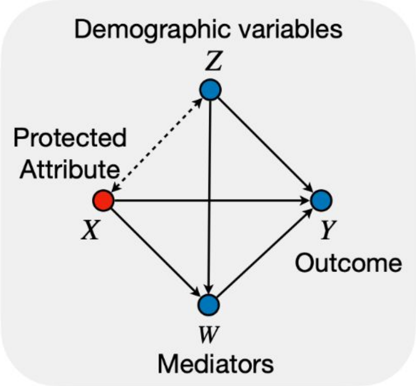

Figure 2: SFM Graph

4.3 Training Methods

Originally, we were trying to train the model data point by data point, but this was ineffective. In doing that, we were effectively training each network separately:

- $\hat{f}_Y$ was learning from real inputs X, W, Z, generated input U_Y, and real label Y.
- Each $\hat{f}_{W_i}$ was learning from real inputs X, Z, generated input U_{W_i}, and the real label W_i.
- Each $\hat{f}_{Z_i}$ was learning only from the generated input U_{XZ_i} and the real label Z_i.
- $\hat{f}_X$ was learning only from the generated inputs U_{XZ} and the real label X.

This meant that while our $\hat{f}_Y$ network was able to generate Y values with relatively high precision, our $\hat{f}_X$ and $\hat{f}_Z$ networks were effectively random guessing.

By the principles of Neural Counterfactual Identification, we don't need to have a model that is completely accurate at each individual point. So long as the resulting probability distribution $\hat{P}(V)$ agrees with the observed distribution $P(V)$, and they can be described by the same causal graph $\mathscr{G}$, the answers to counterfactual queries evaluated on NCM $\hat{M}$ will equal the true counterfactual values within some small margin of error.

Algorithm 1 Basic NCM training pseudo-code

```
procedure TRAIN-NCM(data=D,$\mathscr{G}$)
    $\hat{M} \leftarrow NCM(V, \mathscr{G})$
    Initialize parameters $\theta$
    for each epoch do
        $loss \leftarrow 0$
        for each batch of real datapoints in D do
            $P(V) \leftarrow$ probability distribution of this batch
            $\hat{P}(V) \leftarrow \hat{M}(\theta)$.sample(n)        ▷ where n is the size of this batch
            $loss \leftarrow loss +$ divergence between $P(V)$ and $\hat{P}(V)$
        end for
        Update $\theta$ based on $loss$
    end for
end procedure
```

This was adapted from Algorithm 3 in Xia et al. [2022]. We kept lines 1-4 effectively the same. In Algorithm 3, iterating from k to ℓ affects only the variables V_{z_k} and n_k which are effectively the batch and batch size respectively. Note that typically in causal inference, the notation $V_{variable}$ with respect to $\mathbf{V} = \{endogenous\_variables\}$ indicates an interventional distribution. So $\mathbf{V}_{z_k}$ would typically indicate "variables that have been intervened on to set $Z = z_k$". Typically, real-world data is not interventional; all of our data is observational. However the pseudocode in Xia et al. [2022] was written in this way to accommodate interventional data sources, such as in a randomized control trial. Because we are operating on observational data, it is fine for us to get the batch from a standard Torch DataLoader object instead.

Unlike Algorithm 3, our pseudo-code does not specify a query, nor does it calculate a minimum and maximum $\hat{P}$, loss, or θ. We omit the query because this is just used to track how a given query changes from epoch to epoch. We opted instead to wait until the model was fully trained, and evaluate queries based on this trained model instead. The minimization and maximization, on the other hand, is used to figure out whether a given query is identifiable. Specifically, if $\hat{P}_{max}(query) \neq \hat{P}_{min}(query)$, then the query is not identifiable. If the two probabilities are equal, then they are both equal to the true value $P(query)$. Xia et al. [2022] However we are using a projection onto the standard fairness model and measures which are known to be identifiable from the observational distribution and class of SCMs Ω^{SFM}, so we know that our queries will be identifiable. Therefore we are able to use only one set of parameters for our model, and we know that it will return the correct answer to our query.

4.3.1 Divergence Calculation

To evaluate the divergence for loss calculations during training we needed a divergence function that can work with samples directly. This ruled out many popular divergence metrics like Kullback-Leibler, which measures how a probability distribution differs from a true probability.

Popular sample-based divergence metrics include maximum mean discrepancy (MMD), Wasserstein distance, energy distance, k-nearest neighbors, and classifier-based divergences. Of those, the k-nearest neighbors algorithm does not handle high dimensions well, and Wasserstein is computationally expensive. We implemented an energy based divergence method, MMD, and Jensen-Shannon divergence (which is a type of classifier-based divergence). Of those, MMD was the most discriminatory. For example, on our depression data set, the final testing accuracy was evaluated as follows:

	Depression Data		*Exam Score Data*	
	Gender	**Depression**	**Gender**	**MathScore**
`energy-based`	96.16%	91.32%	95.57%	96.60%
`js-divergence`	95.65%	89.97%	95.08%	96.26%
`MMD`	93.09%	89.98%	92.74%	93.39%

Table 1: Final test scores for X, Y on different divergence metrics.

We used MMD to train because it seemed to be the most discriminatory of the three.

4.4 Counterfactual Sampling

To evaluate a regular probability $P(Y = 1)$, we could sample n points from the trained model and calculate the frequency with which $Y = 1$ in the sample. To evaluate a conditional probability $P(Y = 1|X = x)$, we could sample n points from the model, filter out any samples for which $X \neq x$, and again look for the frequency with which $Y = 1$ in the sample.

Evaluating an intervetional query, something like $P(Y = 1|do(X = x), Z = z)$, you'd follow the same steps as above, except when sampling from the model, you'd force $X = x$ instead of calculating $X = f_X$, and then filter out values for which $Z \neq z$.

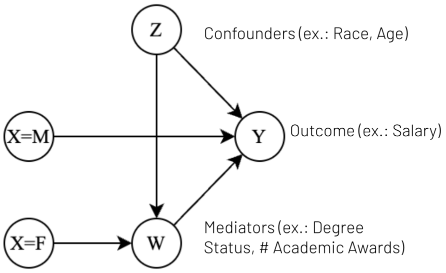

Figure 3: Diagram representing counterfactual $P(Y_{X=M,W_{X=F}})$

A counterfactual query is a bit more complicated. $P(Y_{X=x} = 1|Z = z)$ implies that Y has been intervened on to force $X = x$, but that Z was unaffected by that intervention. So you might calculate $P(Y_{X=x} = 1|Z = z)$ by first sampling the exogenous variables, feeding them into the model to get the full probability distribution $P(V)$, filtering out values for which $Z \neq z$, and then re-sampling the model by feeding it the filtered exogenous variables, and forcing $X = x$. The sampling procedure is detailed in the algorithm below. You could take that sample, then look for the frequency with which $Y = 1$.

Algorithm 2 Basic Counterfactual Sampling

```
procedure SAMPLE-CTF-BASIC(scm, term=CTFTerm, conditions=None, u=None, n=10000)
    U ← Conditioned-U(scm, u, conditions, n)
    sample←scm.sample(u=U,do=term.do-values)
    return sample
end procedure

procedure CONDITIONED-U(scm, u=None, conditions=None, n=10000)
    if u is None then
        U ← scm.pu.sample(n)
    else
        U ← u
        n ← len(u.samples)
    end if
    if conditions is None then return U
    end if
    sample ← scm.sample(u=U)
    indices-to-keep ← set()
    for c in conditions do
        temp-indices ← indices where sample[c]==conditions[c]
        indices-to-keep ← indices-to-keep ∩ temp-indices
    end for
    return U[indices-to-keep]
end procedure
```

Counterfactual queries can be more complex than that, though. The query $P(Y_{X=x_1,W_{x_0}} = 1|X = x_0)$ is considered a "nested counterfactual". One would essentially follow the procedure above to get the filtered U samples, and use those to get a list of samples for W_{x_0}. You'd then use the same U values to get another sample from the model, except instead of calculating $X = f_X$, you'd force $X = x_1$, and instead of calculating $W = f_w$, you'd force W to take on the values from the previous sampled list. In this way, you force $X = x_1$ and $W = W_{x_0}$, even if W_{x_0} varies depending on the provided U value.

Algorithm 3 Counterfactual Sampling

```
procedure SAMPLE-CTF(scm, term=CTFTerm, conditions=None, u=None, n=10000)
    U $\leftarrow$ Conditioned-U(scm, u, conditions, n)
    expanded-dos $\leftarrow$ dict()
    for k in term.do-values do
        if k is nested then          ▷ Calculate nested counterfactual, update related variable
            ctf-sample = Sample-CTF(scm, k.term, U)
            expanded-dos[k.term]$\leftarrow$ctf-sample[k.term])
        else                                    ▷ Force related variable to take a given value
            expanded-dos[k.term]$\leftarrow$term.do-values[k]
        end if
    end for
    sample$\leftarrow$scm.sample(u=U,do=expanded-dos)
    return sample
end procedure
```

Common and useful counterfactual queries include:

- Natural Direct Effect: $NDE_{x_0,x_1}(y) = P(y_{x_1,W_{x_0}}) - P(y_{x_0})$
- Natural Indirect Effect: $NIE_{x_1,x_0}(y) = P(y_{x_1,W_{x_0}}) - P(y_{x_1})$
- Total Effect: $TE_{x_0,x_1}(y) = P(y_{x_1}) - P(y_{x_0}) = NDE - NIE$

The most relevant counterfactual queries related to fairness are the x-specific effects:

- $xDE_{x_0,x_1}(y|x) = P(y_{x_1,W_{x_0}}|x) - P(y_{x_0}|x)$
- $xIE_{x_1,x_0}(y|x) = P(y_{x_1,W_{x_0}}|x) - P(y_{x_1}|x)$
- $xSE_{x_1,x_0}(y) = P(y_{x_0}|x_1) - P(y_{x_0}|x_0)$
- $xTE_{x_0,x_1}(y) = P(y_{x_1}|x) - P(y_{x_0}|x) = xDE - xIE$
- Total Variation: $TV_{x_0,x_1}(y) = xDE_{x_0,x_1}(y|x_0) - xIE_{x_1,x_0}(y|x_0) - xSE_{x_1,x_0}(y)$

5 Results

5.1 Final Y Accuracy

Treating the NCM like a regular supervised learning model. We have compared our NCM model against the best performing models from both Classical ML and Deep Learning. The results on Depression and Exam Scores Dataset is given in Tables 2 and 3 respectively.

Table 2: Accuracy Across Models On Depression Dataset

Models	Accuracy
Decision Tree	82.07%
Random Forest	82.29%
Gradient Boosting	80.71%
LSTM	85.00%
K-Nearest Neighbors	80.15%
Support Vector Machine	84.74%
ResNCM	**89.98%**

Table 3: Accuracy Across Models for Exam Scores Dataset

Models	Accuracy
Decision Tree	90.30%
Random Forest	90.64%
Gradient Boosting	83.03%
Bagging	92.43%
K-Nearest Neighbors	88.90%
Support Vector Machine	93.11%
ResNCM	**93.39%**

5.2 Accuracy for Features!

While standard supervised learning doesn't concern itself accuracies of the features used to train the model. The final goal of Causal Models allow for us to observe the accuracy of predicting the features used in training in addition to the labels. The results on Depression and Exam Scores Dataset is given in Tables 4 and 5 respectively.

Table 4: Accuracy Across Features for Depression Dataset

Models	Accuracy
Gender	96.44%
Depression	90.93%
Academic Pressure	72.86%
Work Pressure	53.48%
CGPA	76.87%
Study Satisfaction	76.17%
Dietary Habits	72.78%
Age	84.65%
Sleep Duration	79.06%
Family History of Mental Illness	91.97%

Table 5: Accuracy Across Features for Exam Scores Dataset

Models	Accuracy
Gender	96.54%
Math Score	95.94%
Ethnic Group	78.93%
Parent Marital Status	76.79%
Is First Child	92.48%
Number of Siblings	62.97%
Transport Means	56.08%
Parent Education	55.90%
Lunch Type	88.82%
Test Prep	91.91%
Practice Sport	67.47%
Weekly Support Hours	71.06%

5.3 Counterfactual Analysis

In legal doctrines, there are two important concepts known as *disparate treatment* and *disparate impact*. Disparate treatment doctrines state that a group should not be treated differently simply for their membership in that group. This is associated with the causal concept of *direct effect* (*DE*): how much a specific attribute directly impacts an outcome. Disparate impact doctrines dictate how much a group may be affected by their membership in that group. For example, students in a given region may have reduced access to tutoring, which in turn may impact their grades. Then the student's region indirectly impacts their grades through the variable `access-to-tutoring`. Disparate impact is associated with the causal concepts of *indirect effect* (*IE*) and *spurious effect* (*SE*). An attribute X may indirectly affect the outcome Y through mediator variables W if X impacts W and W impacts Y. Spurious effect implies that Y is affected by some other attribute Z with which X may be correlated (like how eye color and hair color may be correlated, despite the fact that neither one causes the other – they are both caused by a person's genetic makeup).

Causally, we look for existance of disparate treatment by evaluating $xDE^{sym}(y|x_0) = \frac{1}{2}(xDE_{x_0,x_1}(y|x_0) - xDE_{x_1,x_0}(y|x_0))$. If it is zero, then there is no evidence of disparate treatment. For disparate impact, we look at $xIE^{sym}(y|x_0) = \frac{1}{2}(xIE_{x_0,x_1}(y|x_0) - xIE_{x_1,x_0}(y|x_0))$ and $xSE_{x_0,x_1}(y)$ Plečko et al. [2024].

From our ResNet Neural Causal Model, we got the following results:

	Depression Dataset	**Exam Score Dataset**
TE	4.34%	**11.59%**
NDE	1.98%	**11.35%**
NIE	-2.36%	-0.24%
$xTE(y\|x_1)$	4.79%	**11.43%**
$xDE(y\|x_1)$	2.41%	**12.06%**
$xIE(y\|x_1)$	-2.38%	0.62%
$xSE(y\|x_1)$	-1.90%	-3.15%
$xDE^{sym}(y\|x_1)$	1.98	**11.79%**
$xIE^{sym}(y\|x_1)$	2.36	-0.35%

Table 6: Counterfactual query evaluations from each dataset

In theory we would say that there is evidence of discrimination anytime a relevant value is nonzero, however, in practice we allow for a 5% margin of error. This implies that gender does not have a direct impact on students' depression, but it does have a direct impact on math exam scores.

Interestingly, this wasn't the same with other subjects.

	Reading Score	**Writing Score**
$xDE^{sym}(y\|x_1)$	**-8.60%**	**-5.57%**
$xIE^{sym}(y\|x_1)$	-0.07%	-0.33%
$xSE(y\|x_1)$	**6.21%**	3.07%

Table 7: Counterfactual query evaluations based on different test scores

We tried taking `'ReadingScore'` and `'WritingScore'` out of the mediators when calculating the effects on `'MathScore'`, but the effects didn't change enough to alter the causal implications. Seems gender has a direct effect on students' scores in all subjects. This could have to do with some unobserved mediators. Just because we created a graph with a direct causal connection from X to Y doesn't mean there aren't other attributes outside of our observed mediators W that stand along that path.

6 Conclusion

In this work, we presented ResNCM, a Residual Neural Causal Model, as a robust framework for modeling causal relationships in high-dimensional educational datasets. Our model outperforms both traditional and neural baselines across key prediction tasks and enables counterfactual inference, allowing us to assess the real-world implications of modifying specific attributes in student environments.

The strengths of our approach lie in its ability to jointly model structural equations across a flexible causal graph while supporting expressive reasoning about direct, indirect, and spurious effects. ResNCM demonstrates improved gradient flow and categorical variable handling through residual connections and learned embeddings, making it particularly well-suited for datasets that include both numerical and socio-demographic variables.

However, our findings are not without limitations. Causal interpretations are constrained by the structure of the assumed causal graph, and the presence of unobserved mediators may distort true causal pathways. For instance, although we identified a direct gender effect on math scores, latent variables such as stereotype threat, school climate, or teacher bias—absent from our dataset—may contribute to this relationship.

Looking ahead, we plan to extend the expressiveness of our causal modeling toolkit by integrating generative models such as GANs into the NCM framework. This would enable a deeper exploration of counterfactual distributions and improve performance in settings with limited or imbalanced data, much like our successful integration of ResNet and MLP architectures in this work.

7 Contributions

Aditya: Did Data Preprocessing and Cleaning. Implemented the baselines. Designed the intial MLE-NCM architecture.

Eylam: Designed the ResNCM architecture. Conducted the comparison experiments between the two NCM implementations.

Hanita. Implemented counterfactual queries/sampling, divergence functions, and SCM architecture. Contributed to the final NCM training functions.

References

Hopes B. Student depression dataset, 2022. URL `https://www.kaggle.com/datasets/hopesb/student-depression-dataset`.

Royce Kimmons. Students exam scores dataset, 2023. URL `https://www.kaggle.com/datasets/desalegngeb/students-exam-scores`.

Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018.

Drago Plecko and Elias Bareinboim. Causal fairness analysis. *arXiv preprint arXiv:2207.11385*, 2022.

Drago Plečko, Elias Bareinboim, et al. Causal fairness analysis: a causal toolkit for fair machine learning. *Foundations and Trends® in Machine Learning*, 17(3):304–589, 2024.

Kevin Xia and Elias Bareinboim. Neural causal abstractions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20585–20595, 2024.

Kevin Xia, Kai-Zhan Lee, Yoshua Bengio, and Elias Bareinboim. The causal-neural connection: Expressiveness, learnability, and inference. *Advances in Neural Information Processing Systems*, 34:10823–10836, 2021.

Kevin Xia, Yushu Pan, and Elias Bareinboim. Neural causal models for counterfactual identification and estimation. *arXiv preprint arXiv:2210.00035*, 2022.