Combining Causal Models for More Accurate Abstractions of Neural Networks

Theodora-Mara Pîslar PISLARMARA589@GMAIL.COM
University of Amsterdam

Sara Magliacane SARA.MAGLIACANE@GMAIL.COM
University of Amsterdam

Atticus Geiger ATTICUSG@GMAIL.COM
Pr(Ai)²R Group

Editors: Biwei Huang and Mathias Drton

Abstract

Mechanistic interpretability aims to reverse engineer neural networks by uncovering which high-level algorithms they implement. Causal abstraction provides a precise notion of when a network implements an algorithm, i.e., a causal model of the network contains low-level features that realize the high-level variables in a causal model of the algorithm (Geiger et al., 2024). A typical problem in practical settings is that the algorithm is not an entirely faithful abstraction of the network, meaning it only *partially* captures true reasoning process of a model. We propose a solution where we *combine* different simple high-level models to produce a more faithful representation of the network. Through learning this combination, we can model neural networks as being in different computational states depending on the input provided, which we show is more accurate to GPT 2-small fine-tuned on two toy tasks. We observe a trade-off between the *strength* of an interpretability hypothesis, which we define in terms of the number of inputs explained by the high-level models, and its *faithfulness*, which we define as the interchange intervention accuracy. Our method allows us to modulate between the two, providing the most accurate combination of models that describe the behavior of a neural network given a faithfulness level. The code is available in github.

Keywords: causal abstraction, mechanistic interpretability, causal representation learning

1. Introduction

Great strides have been made in gaining a mechanistic understanding of black box deep learning models using tools from causal mediation (Vig et al., 2020; Finlayson et al., 2021; Mueller et al., 2024) and causal abstraction (Geiger et al., 2020, 2021, 2024; Huang et al., 2024). However, an important area of innovation is quantifying and reasoning about the *faithfulness* of a causal analysis of a deep learning model, i.e., the degree to which the abstract causal model accurately represents the true reasoning process of a model (Lipton, 2018; Jacovi and Goldberg, 2020; Lyu et al., 2022).

In this paper, we explore an approach in which a causal model that is a partially faithful description of a neural network is modified to become more faithful by dynamically changing the causal structure based on what input is provided. We represent algorithms as causal models and define a *combine* operation that creates a causal model with a mechanism that activates different variables based on the input provided, with some inputs activating no intermediate variables. These combined models are more expressive than the original causal models, allowing for more fine-grained hypotheses about the computational process.

However, there is a trade-off between faithfulness and the strength of an interpretability hypothesis. We measure strength by the number of examples that a combination of causal models explains at a given level of faithfulness, i.e., examples assigned to a causal model with internal structure. A causal model with no internal structure has no content and is a trivially faithful hypothesis about network structure; the more interesting structure a causal model has, the more chances there are for inaccuracies. In two experiments with GPT 2-small on arithmetic and boolean logic tasks, we show that combined causal models are able to provide stronger hypotheses at every level of faithfulness.

2. Background

This section covers previous research on causal abstraction and interpretability.

Mechanistic Interpretability. Mechanistic interpretability has converged on analyzing distributed representations, with significant attention to linear subspaces following the linear representation hypothesis (Mikolov et al., 2013; Elhage et al., 2022; Nanda et al., 2023; Park et al., 2023; Jiang et al., 2024). Methods like Distributed Alignment Search (Geiger et al., 2023; Wu et al., 2023) and sparse autoencoders (Bricken et al., 2023; Cunningham et al., 2023) have proven effective in identifying interpretable features, while intervention techniques provide tools for validating interpretability hypotheses, e.g., interchange interventions (Geiger et al., 2020; Vig et al., 2020), path patching (Wang et al., 2023; Goldowsky-Dill et al., 2023), activation patching (Conmy et al., 2023; Zhang and Nanda, 2024; Ghandeharioun et al., 2024), and causal scrubbing (Chan et al., 2022). Circuit analysis has revealed specific computational mechanisms across both visual (Cammarata et al., 2021; Olah et al., 2020) and linguistic domains (Wang et al., 2023; Olsson et al., 2022; Lieberum et al., 2023). Recent theoretical frameworks (Geiger et al., 2024; Mueller et al., 2024) and evaluation methods (Huang et al., 2024) are establishing more rigorous standards for mechanistic explanations.

Faithfulness in Interpretability. Faithfulness is a critical concept in interpretability research, referring to the degree to which an explanation accurately represents the true reasoning process of a model (Jacovi and Goldberg, 2020; Lyu et al., 2022). A faithful interpretation should not only match model behavior, but also reflect the internal decision-making process (Wiegreffe and Pinter, 2019). This is particularly important in high-stakes domains where understanding the model's reasoning is crucial for trust and accountability. However, measuring faithfulness is challenging, as it often requires comparing explanations against ground truth reasoning processes, which are typically unknown for complex models. Despite advances, achieving truly faithful interpretations remains an open challenge in the field of explainable AI (Lipton, 2018).

Causal Models. We adopt the following notation and concepts from Bongers et al. (2021) and Geiger et al. (2024). We define a deterministic causal model $\mathcal{M} = (\Sigma, \{\mathcal{F}_X\}_{X \in \mathbf{V}})$, where $\Sigma = (\mathbf{V}, \mathsf{Val})$ is a signature consisting of a set of variables $\mathbf{V}$ and their corresponding value ranges Val. The mechanisms $\{\mathcal{F}_X\}_{X \in \mathbf{V}}$ assign a value to each variable X as a function of all variables, including itself. We limit ourselves to acyclic models with input variables $\mathbf{X^{In}}$ that depend on no other variables and output variables $\mathbf{X^{Out}}$ on which no other variables depend. The remaining variables are considered intermediate variables. The solution $\mathcal{M}(\mathbf{x})$ of a causal model given an input setting $\mathbf{x}$ is the unique total setting that satisfies the mechanisms for each $X \in \mathbf{V}$, i.e., $\mathsf{Proj}_X(\mathbf{v})$, the projection of $\mathbf{v} \in \mathsf{Val}_{\mathbf{V}}$ onto X, is equal to the mechanism output $\mathcal{F}_X(\mathbf{v})$. While our definition does not explicitly reference a graphical structure, the mechanisms induce a causal ordering $\prec$

among variables, where $Y \prec X$ if there exists a setting $\mathbf{z}$ of variables $\mathbf{Z} = \mathbf{V} \setminus \{Y\}$ and two settings y, y' of Y such that $\mathcal{F}_X(\mathbf{z}, y) \neq \mathcal{F}_X(\mathbf{z}, y')$.

Interventions. Interventions are operations on the mechanisms of a causal model. A hard intervention $\mathbf{i} \in \mathsf{Val}_{\mathbf{I}}$ for $\mathbf{I} \subseteq \mathbf{V}$ replaces the mechanisms $\mathcal{F}_X$ for each $X \in \mathbf{I}$ with constant functions $\mathbf{v} \mapsto \mathsf{Proj}_X(\mathbf{i})$. The causal model resulting from an intervention γ is denoted as $\mathcal{M}_\gamma$.

Interventionals generalize interventions to arbitrary mechanism transformation, i.e., a functional that outputs a new mechanism conditional on the mechanisms of the original model (Geiger et al., 2024). We will not focus on interventionals, though they are necessary for a rigorous treatment of abstraction where high-level variables are aligned to linear subspaces of low-level variables.

Exact Transformation. Exact transformation is a fundamental concept (Rubenstein et al., 2017) that formalizes under which conditions are two causal models $\mathcal{M}, \mathcal{M}^*$ are compatible representations of the same causal phenomena. Exact transformation holds with respect to three maps δ, τ, ω defined on inputs, total settings that assign each variable a value, and interventions, respectively, [1] if for each input $\mathbf{b}$ and intervention $\mathbf{i}$, we have:

$$\tau\Big(\mathcal{M}_{\mathbf{i}}(\mathbf{b})\Big) = \mathcal{M}^*_{\omega(\mathbf{i})}\Big(\delta(\mathbf{b})\Big).$$

This equation holds exactly when every low-level intervention $\mathbf{i}$ applied to the low-level model $\mathcal{M}$ is mapped by τ to the same high-level setting that results from the low-level intervention $\mathbf{i}$ being mapped to a high-level intervention via $\omega(\mathbf{i})$ and then applied to the high-level model $\mathcal{M}^*$. Geiger et al. (2024) generalize this notion to *intervention algebras*, which are sets of interventionals that fix quantities distributed across variables, e.g., a linear subspace of a vector of variables.

Constructive Causal Abstraction. Exact transformation as a general notion is unconstrained; there is no guaranteed relationship between high-level variables and low-level variables. Constructive causal abstraction (Beckers and Halpern, 2019; Beckers et al., 2019; Massidda et al., 2023; Rischel and Weichwald, 2021) is a special case of exact transformation between a low-level model $\mathcal{L}$ and a high-level model $\mathcal{H}$. An alignment $\langle \Pi, \pi \rangle$ assigns each high-level variable $X \in \mathbf{V}_{\mathcal{H}}$ a partition cell $\Pi_X \subseteq \mathbf{V}_{\mathcal{L}}$ of low-level variables and a function π_X for determining the value of X from a setting of Π_X. From an alignment, we can construct a map τ^π from low-level total settings $\mathbf{v}_{\mathcal{L}}$ to high-level total settings:

$$\tau^\pi(\mathbf{v}_{\mathcal{L}}) = \bigcup_{X_{\mathcal{H}} \in \mathbf{V}_{\mathcal{H}}} \pi_{X_{\mathcal{H}}}\big(\mathsf{Proj}_{\Pi_{X_{\mathcal{H}}}}(\mathbf{v}_{\mathcal{L}})\big),$$

where $\mathsf{Proj}_{\Pi_{X_{\mathcal{H}}}}(\mathbf{v}_{\mathcal{L}})$ represents the projection of the low-level total settings to the values for the low-level variables $\pi_{X_{\mathcal{H}}}$. We can also construct a map ω^π from low-level hard interventions to high-level hard interventions where we define $\omega^\pi(\mathbf{x}_{\mathcal{L}}) = \mathbf{x}_{\mathcal{H}}$ iff:

$$\tau^\pi\big(\mathsf{Proj}^{-1}_{\mathbf{V}_{\mathcal{L}}}(\mathbf{x}_{\mathcal{L}})\big) = \mathsf{Proj}^{-1}_{\mathbf{V}_{\mathcal{H}}}(\mathbf{x}_{\mathcal{H}}).$$

A model $\mathcal{H}$ is a constructive abstraction of a model $\mathcal{L}$ iff $\mathcal{H}$ is an exact transformation of $\mathcal{L}$ under $(\delta, \tau^\pi, \omega^\pi)$. Constructive abstraction is not defined on interventionals, and an analysis where high-level variables are aligned to linear subspaces requires an exact transformation of the network to change the basis and the algorithm is a constructive causal abstraction of this transformed model.

1. Previous definitions did not use an input map, we include one here for simplicity's sake.

Applied Causal Abstraction. Causal abstraction has been used to analyze weather patterns (Chalupka et al., 2016), human brains (Dubois et al., 2020a,b), physical systems (Kekic et al., 2023), batteries (Zennaro et al., 2023), epidemics (Dyer et al., 2023), and deep learning models (Chalupka et al., 2015; Geiger et al., 2021; Hu and Tian, 2022; Geiger et al., 2023; Wu et al., 2023).

3. Mechanistic Interpretability via Causal Abstraction

The core of mechanistic interpretability is reverse engineering the algorithms implemented by a neural network to solve a task. We operationalize a mechanistic interpretability hypothesis about internal structure by representing neural networks and algorithms both as causal models and then understanding implementation as a type of exact transformation. However, to uncover the structure of a neural network, we need to *featurizes* hidden representations. We use the high-level causal model as a source of supervision to learn a bijective exact transformation that creates groups of orthogonal linear features corresponding to variables in a high-level neural network.

Interchange Interventions. The theory of causal abstraction does not specify *which* interventions should be evaluated. Geiger et al. (2021) argue for using interventions that fix variables to the values they would have if some counterfactual input were provided, as such low-level counterfactual states have meanings determined fully by the input and the high-level model. Given a model $\mathcal{M}$ with counterfactual input $\mathbf{s}$ and target variables $\mathbf{X}$, a single source interchange intervention is $\mathsf{IntInv}(\mathcal{M}, \mathbf{s}, \mathbf{X}) = \mathsf{Proj}_{\mathbf{X}}(\mathcal{M}(\mathbf{s}))$, the value that $\mathbf{X}$ takes on when $\mathbf{s}$ is input.

Distributed interchange interventions (DII) generalize standard interchange interventions to target quantities that are distributed across several causal variables. This is crucial for analyzing complex neural networks where individual neurons often participate in multiple conceptual roles. A bijective function is applied before the interchange intervention and the inverse is applied after. A distributed interchange intervention is an interventional, i.e., rather than fixing variables to constant values, the variable mechanisms are replaced with a new set of mechanisms.

For our purposes, the bijective function is a rotation that allows interventions on the dimensions of a new basis. The hyperparameter k determines the number of dimensions we will intervene on. Given a model $\mathcal{M}$, counterfactual input $\mathbf{s}$, hidden representation $\mathbf{H} \in \mathbb{R}^n$, and a matrix with k orthogonal columns $\mathbf{R} \in \mathbb{R}^{n \times k}$, the distributed interchange intervention is the intervention $\mathsf{DistIntInv}(\mathcal{M}, \mathbf{s}, \mathbf{H}, \mathbf{R})$ that fixes the linear subspace spanned by the columns of $\mathbf{R}$ to the value they take for counterfactual input $\mathbf{s}$. The mechanism of $\mathcal{M}_{\mathsf{DistIntInv}(\mathcal{M},\mathbf{s},\mathbf{H},\mathbf{R})}$ for the variables $\mathbf{H}$ is

$$\mathbf{v} \mapsto \mathcal{F}_{\mathbf{H}}(\mathbf{v}) + \mathbf{R}^T \left(\mathbf{R}\Big(\mathcal{F}_{\mathbf{H}}\big(\mathcal{M}(\mathbf{s})\big)\Big) - \mathbf{R}\big(\mathcal{F}_{\mathbf{H}}(\mathbf{v})\big) \right) \tag{1}$$

This new mechanism maps the total setting $\mathbf{v}$ to the original mechanism, except the subspace spanned by $\mathbf{R}$ is erased by subtracting $\mathbf{R}\big(\mathcal{F}_{\mathbf{H}}(\mathbf{v})\big)$ and then fixed to be $\mathbf{R}\Big(\mathcal{F}_{\mathbf{H}}\big(\mathcal{M}(\mathbf{s})\big)\Big)$, the value that the subspace would take on when the counterfactual input $\mathbf{s}$ is run through the model $\mathcal{M}$.

Distributed Alignment Search. Distributed Alignment Search (Geiger et al. 2023; DAS) is a method for aligning linear subspaces of a hidden representation with high-level variables. DAS optimizes an orthogonal matrix $\mathbf{R}$ using distributed interchange interventions. Given a fixed alignment, define a loss function for each high-level variable X and low-level variables Π_X with rotation

matrix $\mathbf{R}_X$. Given a base input $\mathbf{b}$, counterfactual input $\mathbf{s}$, and high-level model $\mathcal{H}$, the loss is

$$\mathcal{L}_{\mathsf{DAS}} = \sum \mathsf{CE}\big(\mathcal{L}_{\mathsf{DistIntInv}(\mathcal{L},\mathbf{s},\Pi_X,\mathbf{R}_X)}(\mathbf{b}), \mathcal{H}_{\mathsf{IntInv}(\mathcal{H},\mathbf{s},X)}(\mathbf{b})\big) \tag{2}$$

where CE is the cross-entropy loss.

Approximate Transformation and Interchange Intervention Accuracy. While exact transformation provides a strict criterion for abstraction, in practice we quantify the degree to which a high-level model approximates a low-level model. Geiger et al. (2024) generalize exact transformation to approximate transformation by introducing a similarity function Sim between total settings, a probability distribution $\mathbb{P}$ over interventions, and a statistic $\mathbb{S}$ to aggregate similarity scores. Interchange Intervention Accuracy (IIA) is a specific instance of approximate abstraction that measures the proportion of interchange interventions where the low-level and high-level models produce the same output. IIA for a single high-level variable X over counterfactual dataset $\mathcal{D}$ is defined as:

$$\mathsf{IIA}(\mathcal{H},\mathcal{L},\delta,\tau,\omega,\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{\mathbf{b},\mathbf{s}\in\mathcal{D}} \mathbb{1}\big[\mathcal{L}_{\mathsf{DistIntInv}(\mathcal{L},\mathbf{s},\Pi_X,\mathbf{R}_X)}(\mathbf{b}) = \mathcal{H}_{\mathsf{IntInv}(\mathcal{H},\mathbf{s},X)}(\mathbf{b})\big] \tag{3}$$

Interchange intervention accuracy will serve as our faithfulness metric quantifying the degree to which a high-level causal model $\mathcal{H}$ is a causal abstraction of a low-level neural network $\mathcal{L}$. Appendix C.2 shows a step-by-step example of how a counterfactual dataset is constructed.

4. Preliminary Interpretability Analysis on a Simple Arithmetic Task

We will motivate the approach in this work by demonstrating the issues of partially faithful abstractions. Our running example is the task of summing three numbers X, Y, and Z that vary from 1 to 10. We fine-tune GPT 2-small to perfectly solve the task, but the fundamental problem of interpretability is that deep learning models are black boxes and we do not know *how* the model solves the task. However, this is a simple task and we can easily enumerate some algorithms that solve it.[2]

4.1. Interpretability Hypotheses

We consider five different states that GPT 2-small might be in when adding three numbers X, Y, Z. One initial state where X, Y, and Z have not been summed at all. three partial states where X and Y have been summed, Y and Z have been summed, or X and Z have been summed, and one final state: X and Y and Z have all been summed. We articulate each of these states as interpretability hypotheses using high-level causal models that will be aligned to the GPT 2-small model trained to perform addition.

A layer of GPT 2-small has not yet summed X, Y, and Z if there are three components that separately represent X, Y, and Z. To localize each variable, we define models $\mathcal{M}^X, \mathcal{M}^Y, \mathcal{M}^Z$ with intermediate variable P representing the processed input and O representing the output:

$$\mathcal{M}^X\colon \quad \mathcal{F}_P(X) = X, \qquad\qquad \mathcal{F}_O(P,Y,Z) = P + Y + Z \tag{4}$$

2. Obviously, there are an infinite number of algorithms that solve any input-output task, as useless computational structure can be added ad infinitum. Nonetheless, a simple input-output task makes this space easier to think about and search through.

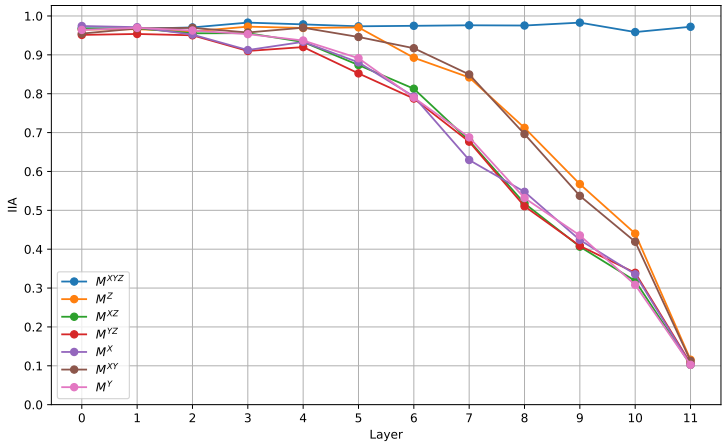


Figure 1: Interchange Intervention Accuracy (IIA) for the intermediate variables P of different high-level causal models across GPT layers when using a 256-dimensional alignment subspace. Early layers (1-4) show high accuracy for models representing separate variables ($\mathcal{M}^X$, $\mathcal{M}^Y$, $\mathcal{M}^Z$), indicating no summation has occurred. The complete summation model ($\mathcal{M}^{XYZ}$) maintains perfect accuracy across all layers. From layers 5-11, there is a gradual transition where partial summation models ($\mathcal{M}^{XY}$, $\mathcal{M}^{YZ}$, $\mathcal{M}^{XZ}$) show intermediate accuracy, suggesting the network does not transition between computational states in discrete steps.

Similarly define $\mathcal{M}^Y$ and $\mathcal{M}^Z$. Also, we can define models where pairs of variables have been summed into an intermediate representation P, with the remaining variable to be added for the final sum in the output O:

$$\mathcal{M}^{XY}: \quad \mathcal{F}_P(X,Y) = X + Y, \qquad \mathcal{F}_O(P,Z) = P + Z \tag{5}$$

Similarly define $\mathcal{M}^{YZ}$ and $\mathcal{M}^{XZ}$. These three models represent intermediate computational states where exactly two of the input variables have been summed, while the third remains separate. Each model captures a different possible ordering of operations, reflecting different paths through the computation. Lastly, define the model where all three variables are summed at the same time:

$$\mathcal{M}^{XYZ}: \quad \mathcal{F}_P(X,Y,Z) = X + Y + Z, \qquad \mathcal{F}_O(P) = P \tag{6}$$

This model represents the final computational state where all three numbers have been added together. The intermediate variable P holds the complete sum, and the output mechanism simply returns this value unchanged.

Each model captures a distinct hypothesis about the internal state of the network at a given layer, allowing us to track how the computation progresses through these states. Previous approaches would evaluate the degree to which a layer of GPT 2-small adheres to each hypothesis.

4.2. Distributed Alignment Search Results By Layer

For each causal model, we localize the intermediate variable P to a k-dimensional linear subspace of the full residual stream of GPT 2-small at a fixed layer L, i.e., the output of the Lth transformer block with in $\mathbb{R}^{N \times d}$ where $N = 6$ is the number of tokens (X+Y+Z=) and $d = 768$ is the model dimensionality. We use the implementation of distributed alignment search (DAS) from the `pyvene` library (Wu et al., 2024). We explore different values $k \in \{64, 128, 256\}$, and find stable results across these three settings. More information about the hyperparameters used can be found in Appendix D. The results for $k = 256$ in terms of the Interchange Intervention Accuracy (IIA) for each of the 12 layers are shown in Figure 1.

The Partial State $\mathcal{M}^{XY}$ is Most Accurate. $\mathcal{M}^{XY}$ is more accurate for IIA than $\mathcal{M}^{YZ}$ and $\mathcal{M}^{YZ}$; the asymmetry in this solution is likely due to the causal attention of GPT 2-small, i.e., the model can look at X, but not Z, when it processes Y.

Early and Late Layers are in Stable States. In early layers of GPT 2-small, the model is in a state where no summation has occurred. The models $\mathcal{M}^{X}$, $\mathcal{M}^{Y}$, and $\mathcal{M}^{Z}$ can each have their intermediate variable P localized to the output of the transformer block up through layer 4 of GPT 2-small. At layer 11, GPT 2-small has completely summed the three numbers and no model can have its intermediate variable localized other than $\mathcal{M}^{XYZ}$. The model $\mathcal{M}^{XYZ}$ is perfectly accurate at all layers, which tells us that a 256 dimensional subspace of the 5×768 dimension residual stream is sufficient to mediate the causal effect of inputs on outputs.

For Many Intermediate Layers, the Model Partially Represents Multiple Quantities. Between layers 4 and 11, there is a gradual transition. The $\mathcal{M}^{XY}$ and $\mathcal{M}^{Z}$ are more accurate in terms of IIA than the other models, but everything other than $\mathcal{M}^{XYZ}$ (the trivial model) is only a partially accurate representation of what is going on at these layers. Based on these findings, we know that for each layer the model represents different quantities for different inputs. Our proposal is then to combine multiple causal models in order to form a new model that activates different causal processes based on the input provided to the model. This will allow us to construct a more faithful description of the network.

5. Combining High-Level Abstract Models For More Faithful Abstractions of LLMs

The approximate abstraction results in the previous section with partial faithfulness are difficult to interpret. If 70% of interchange interventions are successful for a given alignment, does it make sense to think of the analysis as mostly successful, or could the high-level model be a completely misleading explanation of what is going on in the network? Early work attempted to avoid this issue by identifying the largest subset of the input space for which abstraction relation holds perfectly (Geiger et al., 2020, 2021). This specific evaluation was too rigid, as a single failed interchange intervention results in total failure. However, the general approach of adding nuance to the hypothesis in order to increase accuracy is promising.

In this work, we propose a flexible framework for combining and weakening interpretability hypotheses. The core idea is that subsets of model inputs are assigned to different high-level models which describe different points of a computational process. These high-level models are then *combined* to form a more nuanced hypothesis that reflects the fact that the low-level network does not discretely transition between stages of computation, with sharp steps.

Definition 1 (Combined Causal Models) *Let $\mathcal{M}^1, \ldots, \mathcal{M}^k$ be causal models with identical input variables $\mathbf{X^{In}}$ and output variables $\mathbf{X^{Out}}$, but distinct intermediate variables $\mathbf{V}_1, \ldots, \mathbf{V}_k$, respectively. Define the combined $\mathcal{M}^*$ of causal models $\mathcal{M}^1, \ldots, \mathcal{M}^k$ with input space partition $\Delta_1, \ldots \Delta_k$ as follows. The intermediate variables of $\mathcal{M}^*$ are the union of all intermediate variables $\mathbf{V}^* = \bigcup_{j=1}^{k} \mathbf{V}_j$ and the mechanism for $X \in \mathbf{V}^*$ is*

$$\mathcal{F}^*_X(\mathbf{v}^*) = \begin{cases} \mathcal{F}^j_X(\mathsf{Proj}_{\mathbf{V}_j}(\mathbf{v}^*)) & \text{if } \mathsf{Proj}_{\mathbf{X^{In}}}(\mathbf{v}^*) \in \Delta_j \\ \emptyset & \textit{otherwise}, \end{cases}$$

where $\mathbf{v}^* \in \mathsf{Val}_{\mathbf{V}^*}$. The mechanisms of the combined causal model $\mathcal{M}^*$ are a piecewise combination of the mechanisms from the uncombined models $\mathcal{M}^1, \ldots, \mathcal{M}^k$. If an input $\mathbf{x} \in \mathsf{Val}_{\mathbf{X^{In}}}$ is in partition Δ_j, then the variables $\mathbf{V}_j$ are activated and the variables from other models are set to a null value $\emptyset$.

Let us demonstrate the combination of causal models with a concrete example from our arithmetic task. Consider combining three models: $\mathcal{M}^{XY}$ (sum X, Y first), $\mathcal{M}^{YZ}$ (sum Y, Z first), and $\mathcal{M}^{XYZ}$ (direct sum), with input space partitioned by the magnitude of X: $\Delta_1 = \{(x, y, z) : x \leq 3\}$, $\Delta_2 = \{(x, y, z) : 3 < x \leq 6\}$, and $\Delta_3 = \{(x, y, z) : x > 6\}$. The intermediate variables are P_{XY}, P_{YZ}, and P_{XYZ} with mechanisms defined as follows:

$$\mathcal{F}_{P_{XY}}(x, y) = \begin{cases} x + y & \text{if } (x, y, z) \in \Delta_1 \\ \emptyset & \text{otherwise} \end{cases} \qquad \mathcal{F}_{P_{YZ}}(y, z) = \begin{cases} y + z & \text{if } (x, y, z) \in \Delta_2 \\ \emptyset & \text{otherwise} \end{cases}$$

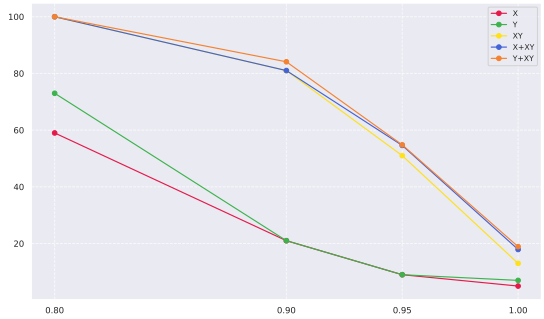
$$\mathcal{F}_{P_{XYZ}}(x, y, z) = \begin{cases} x + y + z & \text{if } (x, y, z) \in \Delta_3 \\ \emptyset & \text{otherwise} \end{cases}$$

The output mechanism combines these intermediate computations:

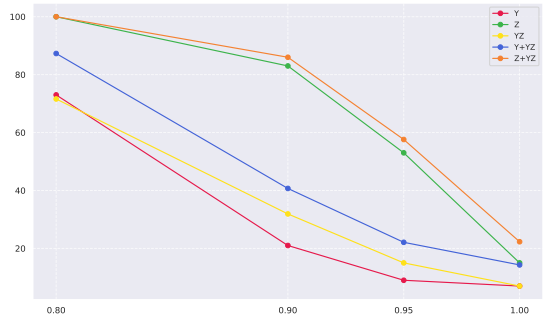
$$\mathcal{F}_O(x, y, z, p_{xy}, p_{yz}, p_{xyz}) = \begin{cases} p_{xy} + z & \text{if } (x, y, z) \in \Delta_1 \\ x + p_{yz} & \text{if } (x, y, z) \in \Delta_2 \\ p_{xyz} & \text{if } (x, y, z) \in \Delta_3 \end{cases}$$

This combined model expresses the hypothesis that the network employs different addition strategies based on the magnitude of the input variable X: step-by-step addition for small numbers ($X \leq 3$), a different sequential strategy for medium-sized numbers ($3 < X \leq 6$), and direct computation for large numbers ($X > 6$).

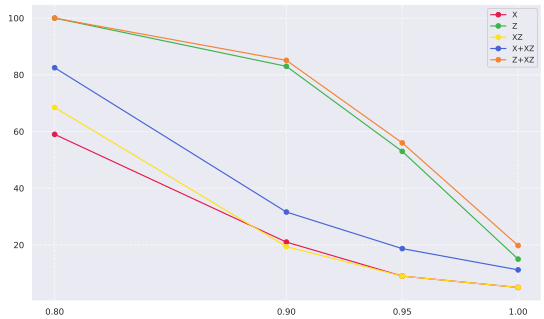
The Faithfulness-Strength Trade-Off. The approach of combining models allows us to precisely quantify the trade-off between the strength of an interpretability hypothesis and its faithfulness to the underlying neural network. A stronger hypothesis makes more specific claims about the computational process by including more inputs in the partition cells for models with internal structure. At one extreme, we can assign all inputs to the model that makes no claims about intermediate computation (in the arithmetic task $\mathcal{M}^{XYZ}$), achieving perfect faithfulness but providing no insight into the network's operation. We will refer to this model as the *trivial model*, since it is trivially perfectly accurate and, at the same time, still completely a black box. At the other extreme, we can assign all inputs to a single causal model, making a strong claim at the risk of inaccuracies. We will



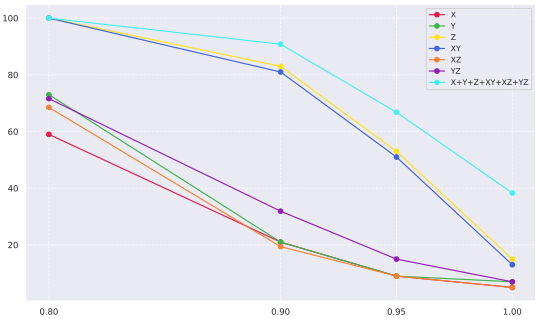
(*a*) Comparison of X and Y term combinations



(*b*) Comparison of Y and Z term combinations



(*c*) Comparison of X and Z term combinations



(*d*) Overall comparison of all models

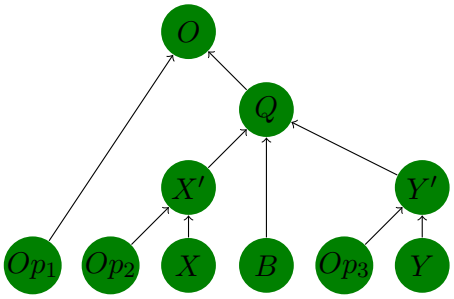
Figure 2: Analysis of layer 7 causal models in fine-tuned GPT model on arithmetic tasks. The x-axis quantifies faithfulness with the interchange intervention accuracy achieved for the hypothesis. The y-axis quantifies strength as the proportion of inputs assigned to a non-trivial high-level causal model. Results compare the performance of individual causal models versus their combinations across different faithfulness thresholds. Combined models demonstrate stronger hypotheses at high faithfulness levels (0.9-1.0), while all models converge in performance at lower faithfulness thresholds (0.8).

evaluate our proposed combined models on their ability to provide hypotheses that are both stronger and more faithful than any of the original causal models.

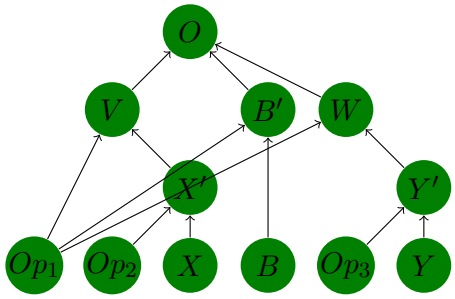
Faithfulness is quantified as interchange intervention accuracy (See Section 2), i.e., degree to which neural network features play the same causal role as aligned high-level variables. We quantify strength as the proportion of inputs not assigned to a trivial model.

Definition 2 (Model Strength) *Let $\mathcal{M}^*$ be a combined model of causal models $\mathcal{M}^1, \ldots, \mathcal{M}^k$ with input space partition $\Delta_1, \ldots, \Delta_k$. Without loss of generality, let $\mathcal{M}^k$ be the trivial model. The* strength *of $\mathcal{M}^*$ is the proportion of inputs not assigned to the trivial model* $1 - \frac{|\Delta_k|}{|\mathsf{Val}_{\mathbf{X}\mathbf{In}}|}$.

Aligning Combined Models. In Section 4, we described experiments where the intermediate variable from each causal model is aligned with a low dimensional linear subspace of the transformer residual stream. To align the variables in a combined model, we simply take the alignment learned for each of the original models.



(*a*) Evaluate the inner logic first and then apply the outer operator.



(*b*) Use De Morgan's laws, i.e., an input $\neg(\neg X \wedge Y)$ is converted to $X \vee \neg Y$.

Figure 3: Two hypotheses for solving the boolean task.

Evaluation Graphs. We need to assign as many inputs to the partition cells $\Delta_1, \ldots, \Delta_{k-1}$ while remaining faithful to the low-level neural network. We precompute the interchange intervention accuracies for each model $\mathcal{M}^1, \ldots \mathcal{M}^{k-1}$ in the form of *evaluation graphs* G^1, G^{k-1}. In this weighted graph, the nodes are inputs, meaning there are 1000 nodes for every possible $(x, y, z) \in \mathsf{Val}_{\mathbf{X}_{\mathbf{In}}}$. For inputs (x, y, z) and (x', y', z'), there are two interchange interventions depending on which input is the base and which is the source. The edge in G^j is weighted 0, 0.5, or 1 according to the intervention interchange accuracy for model $\mathcal{M}^j$ on the two inputs. $\mathcal{M}^j$ is a perfect abstraction of a neural network when its corresponding evaluation graph G^j is a complete graph where all edges have weight 1. The pseudocode for obtaining an evaluation graph and a step-by-step example is available in Appendix A.1.

Greedily Constructing Input Space Partitions. We find the input partitions for a combined causal model using a greedy approach. The algorithm takes evaluation graphs $\{G^1, ..., G^k\}$ as an input and aims to partition the input space $\mathcal{X}$ across a set of candidate models $\{\mathcal{M}^1, ..., \mathcal{M}^k\}$, ensuring each model meets a minimum faithfulness threshold λ on its assigned inputs.

The algorithm is as follows. First, for each candidate model $\mathcal{M}^j$, we greedily identify the largest possible set of currently unassigned inputs on which $\mathcal{M}^j$ achieves the faithfulness threshold λ.

1. The nodes of G^j are sorted from highest degree to lowest degree.
2. Nodes are added to a subgraph S^j until the next node would result in an interchange intervention accuracy, i.e., the average edge weight of S^j, that exceeds λ.
3. The best model is $\mathcal{M}^j$ where $j = \mathsf{argmax}_j(|S^j|)$ is assigned input space Δ_j corresponding to the nodes of subgraph S^j.

Then, the nodes from S^j are removed from the graph, and we repeat the process until either all inputs have been assigned or no remaining model can faithfully handle any of the unassigned inputs. The algorithm returns both the set of selected models and their corresponding input partitions. The pseudocode and a step-by-step example can be found in Appendix A.2.

6. Experimental results

We demonstrate the effectiveness of our combination approach through experiments on GPT 2-small fine-tuned on two toy tasks. While these tasks are deliberately simple to allow for clear analysis and

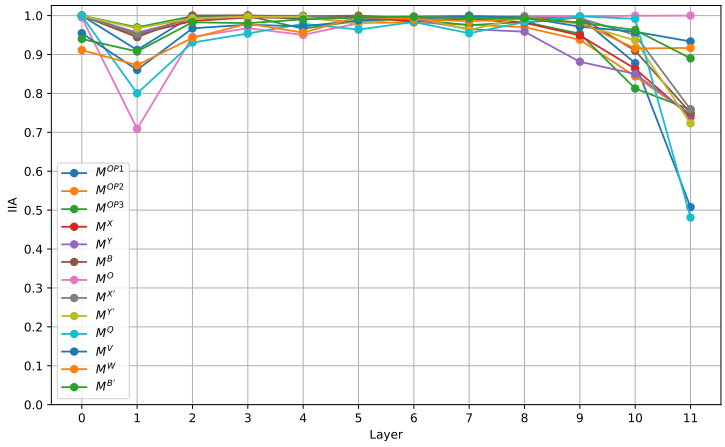


Figure 4: Interchange Intervention Accuracy (IIA) for the intermediate variables P of different high-level causal models of the boolean logic task across the layers of GPT 2-small when searching within 256-dimensional subspaces of the neural representations.

validation, they serve as important proofs of concept. By showing that our method can successfully combine causal models on these controlled examples, we establish a foundation for applying these techniques to more complex architectures and real-world tasks. Our results consistently show that combining causal models leads to more robust and complete explanations of model behavior compared to analyzing individual causal models in isolation.

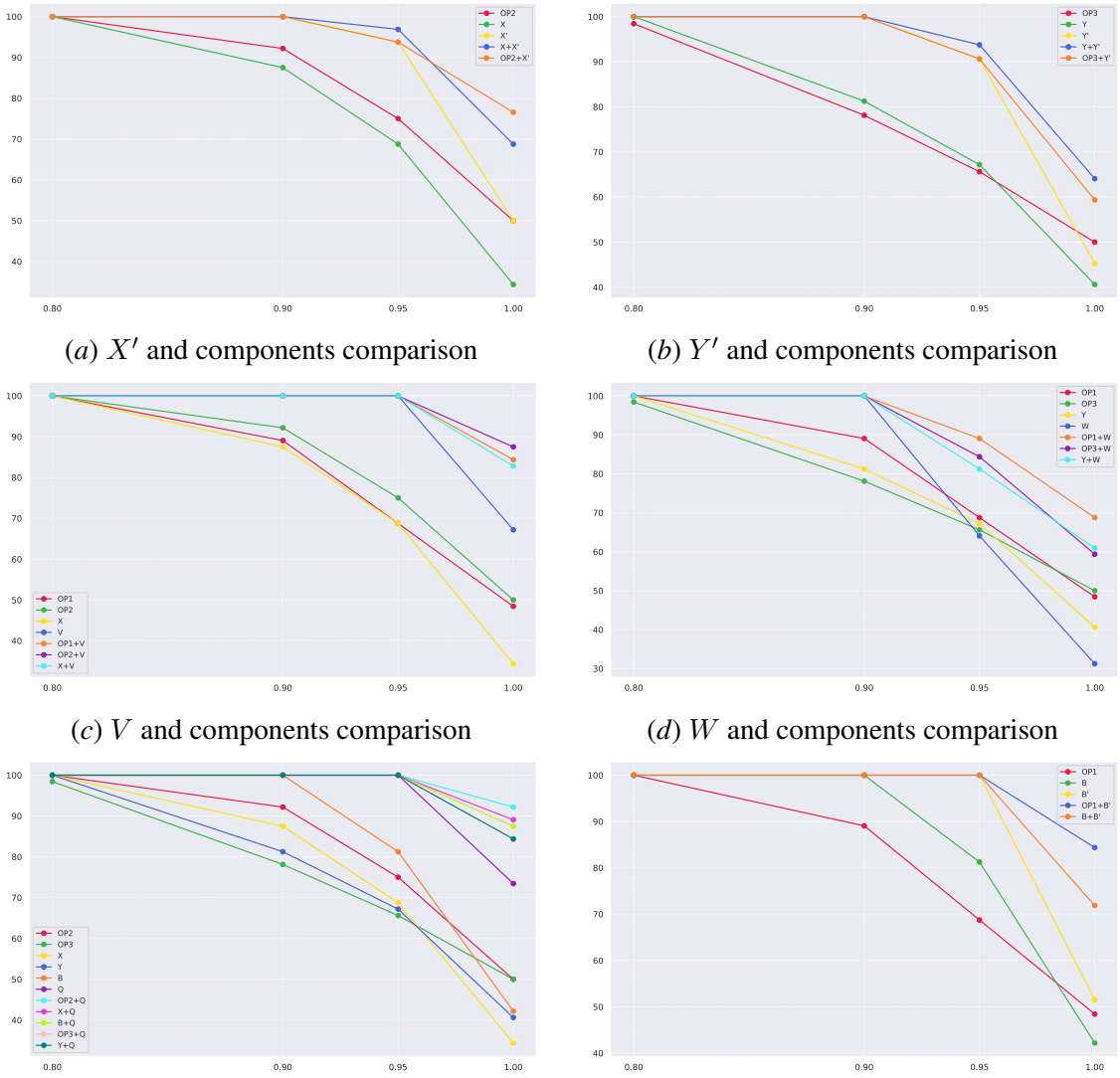
6.1. Arithmetic Task Results

We will now use our algorithm to analyze layer 7 from the GPT 2-small model we fine-tuned on the arithmetic task, as described in Section 4. For each faithfulness threshold (1, 0.95, 0.9, 0.8), we greedily assign inputs to the available causal models until the threshold is met. In Figure 2, we compare uncombined and combined causal models.

Combined Models Provide Stronger Hypotheses at High Levels of Faithfulness. In Figure 2(*a*), we can see that the combination of $\mathcal{M}^Y$ or $\mathcal{M}^X$ with $\mathcal{M}^{Y+XY}$ is able to provide a stronger hypothesis than any of those models alone. Similarly in Figure 2(*b*) the combination of $\mathcal{M}^Z$ and $\mathcal{M}^{YZ}$ is stronger and in Figure 2(*c*) the combination of $\mathcal{M}^Z$ and $\mathcal{M}^{XZ}$ is stronger. In Figure 2(*d*), we can see that the full combination is the strongest overall for high-levels of faithfulness.

6.2. Boolean Logic Task

Our second task is basic boolean logic with unary and binary operators. The task can be encoded into a prompt with the form $OP_1\ (OP_2(X)\ B\ OP_3(Y))\ =$, where boolean variables $X, Y \in \{\texttt{True}, \texttt{False}\}$, and the unary operations $OP_1, OP_2, OP_3 \in \{\neg, \texttt{I}\}$, where $\neg$ represents the unary negation operator, while $\texttt{I}$ is a unary identity operator. The B is a boolean operator that can be



(a) X' and components comparison

(b) Y' and components comparison

(c) V and components comparison

(d) W and components comparison

(e) Q and components comparison

(f) B' and components comparison

Figure 5: Analysis of layer 10 causal models in fine-tuned GPT2-small model on boolean logic tasks. Results compare the performance of individual causal models versus their combinations across different faithfulness thresholds. Combined models demonstrate stronger hypotheses at high faithfulness levels (0.9-1.0).

either $\wedge$ (the AND operator) or $\vee$ (the OR operator). There are 64 possible inputs. We again fine-tune GPT2-small to perfectly predict the truth value of the input. Once again, it is unknown *how* the model solves this task. However, we can come up with two intuitive solutions, and further split them into smaller states GPT2-small might encode within its neural representations.

Interpretability Hypotheses. A direct solution is to evaluate the expressions $X' := OP_2(X)$ and $Y' := OP_3(Y)$, then evaluate $Q := X'BY'$, before finally evaluating $O := OP_1(Q)$. This

algorithm is shown in Figure 3a. The other approach is to use De Morgan's laws. In this case, the unary operator OP_3 is applied to X', B, and Y', to form the variable V, B', and W, respectively, where $OP_3(\wedge) = \vee$ and $OP_3(\vee) = \wedge$. The causal model is shown in Figure 3b. For each of these variables, we define a model with a single intermediate variable just as we did the arithmetic task. Explicit function definitions for each of the models refer to Appendix E.

Distributed Alignment Search Results By Layer. Figure 4 shows the evaluation of the boolean logic task in terms of IIA when searching for alignments within 256-dimensional linear subspaces for each of the 12 layers of GPT2-small. The prompt for the boolean task has a fixed size of 15 tokens, each encoded in a 768 dimensional space. This means that for the layer L, the Lth transformer block has a representation $\mathbb{R}^{15\times768}$. Observe that in layer 10, we begin to see a differentiation between the models, e.g., $\mathcal{M}^X$ has lower IIA compared to the ones that are dependent on more variables like $\mathcal{M}^{X'}$. Based on this, we focus our analysis on layer 10 of the model.

Combining Models For the Boolean Logic Task We analyze layer 10 of GPT2-small fine-tuned on the boolean logic task. Similar to the previous experiment on the arithmetic task, for each faithfulness threshold (1,0.95, 0.9, 0.8). Figure 5 shows the comparison between combined variables and individual ones. We notice how across faithfulness levels for each of the combined variables models, the strength of a combined candidate model is higher than the result when considering each model individually. This stays consistent with the results from the previous experiments on the arithmetic task, even though the input space is much smaller (only 64 inputs compared to 1000 inputs). In Figure 5(*a*), combining $\mathcal{M}^X$ or $\mathcal{M}^{OP_2}$ with $\mathcal{M}^{X'}$ resulting in $\mathcal{M}^{X+X'}$ or $\mathcal{M}^{OP_2+X'}$ can interpret a bigger input space compared to each of those models alone. For example, without combining the models, only 50% or lower of data is interpretable with 100% faithfulness, but when using a combined model, 75% of data is interpretable with 100% faithfulness. Also, in Figure 5(*c*) we have that $\mathcal{M}^{OP2+V}$ abstracts 88% of the input space with 100% faithfulness, whereas the uncombined models only cover below 50% of the input space. These trends can be observed across plots for different combined models. At 0.8 faithfulness, all models uncombined and combined models are equal at full strength.

7. Conclusion.

This work offers an approach to a fundamental challenge in mechanistic interpretability: the trade-off between hypothesis strength and faithfulness. By introducing a framework for combining causal models that activate different computational processes based on input, we enable more nuanced interpretations of neural computation. Our experiments on arithmetic and Boolean logic tasks with GPT 2-small demonstrate that combined models can provide stronger interpretability hypotheses while maintaining high faithfulness compared to uncombined models, particularly at high faithfulness thresholds (0.9-1.0). While we focused on toy tasks for clear analysis and validation, this framework could be extended to more complex tasks, integrated with other mechanistic interpretability methods, and enhanced with more sophisticated optimization objectives for constructing input space partitions.

Acknowledgments

We thank SURF for the support in using the National Supercomputer Snellius. AG was supported by grants from Open Philanthropy.

References

Sander Beckers and Joseph Halpern. Abstracting causal models. In *AAAI Conference on Artificial Intelligence*, 2019.

Sander Beckers, Frederick Eberhardt, and Joseph Y. Halpern. Approximate causal abstractions. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, 2019.

Stephan Bongers, Patrick Forré, Jonas Peters, and Joris M. Mooij. Foundations of structural causal models with cycles and latent variables. *The Annals of Statistics*, 49(5):2885–2915, 2021.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Nick Cammarata, Gabriel Goh, Shan Carter, Chelsea Voss, Ludwig Schubert, and Chris Olah. Curve circuits. *Distill*, 2021. doi: 10.23915/distill.00024.006. https://distill.pub/2020/circuits/curve-circuits.

Krzysztof Chalupka, Pietro Perona, and Frederick Eberhardt. Visual causal feature learning. In Marina Meila and Tom Heskes, editors, *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, UAI 2015, July 12-16, 2015, Amsterdam, The Netherlands*, pages 181–190. AUAI Press, 2015. URL http://auai.org/uai2015/proceedings/papers/109.pdf.

Krzysztof Chalupka, Tobias Bischoff, Frederick Eberhardt, and Pietro Perona. Unsupervised discovery of el nino using causal feature learning on microlevel climate data. In Alexander T. Ihler and Dominik Janzing, editors, *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, UAI 2016, June 25-29, 2016, New York City, NY, USA*. AUAI Press, 2016. URL http://auai.org/uai2016/proceedings/papers/11.pdf.

Lawrence Chan, Adrià Garriga-Alonso, Nicholas Goldwosky-Dill, Ryan Greenblatt, Jenny Nitishinskaya, Ansh Radhakrishnan, Buck Shlegeris, and Nate Thomas. Causal scrubbing, a method for rigorously testing interpretability hypotheses. *AI Alignment Forum*, 2022. https://www.alignmentforum.org/posts/JvZhhzycHu2Yd57RN/causal-scrubbing-a-method-for-rigorously-testing.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine,

editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Abstract-Conference.html.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *CoRR*, abs/2309.08600, 2023. doi: 10.48550/ARXIV.2309.08600. URL https://doi.org/10.48550/arXiv.2309.08600.

Julien Dubois, Frederick Eberhardt, Lynn K. Paul, and Ralph Adolphs. Personality beyond taxonomy. *Nature human behaviour*, 4 11:1110–1117, 2020a.

Julien Dubois, Hiroyuki Oya, Julian Michael Tyszka, Matthew A. Howard, Frederick Eberhardt, and Ralph Adolphs. Causal mapping of emotion networks in the human brain: Framework and initial findings. *Neuropsychologia*, 145, 2020b.

Joel Dyer, Nicholas Bishop, Yorgos Felekis, Fabio Massimo Zennaro, Anisoara Calinescu, Theodoros Damoulas, and Michael J. Wooldridge. Interventionally consistent surrogates for agent-based simulators. *CoRR*, abs/2312.11158, 2023. doi: 10.48550/ARXIV.2312.11158. URL https://doi.org/10.48550/arXiv.2312.11158.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022.

Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. Causal analysis of syntactic agreement mechanisms in neural language models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1828–1843, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.144. URL https://aclanthology.org/2021.acl-long.144.

Atticus Geiger, Kyle Richardson, and Christopher Potts. Neural natural language inference models partially embed theories of lexical entailment and negation. In Afra Alishahi, Yonatan Belinkov, Grzegorz Chrupała, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad, editors, *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.16. URL https://aclanthology.org/2020.blackboxnlp-1.16.

Atticus Geiger, Hanson Lu, Thomas F Icard, and Christopher Potts. Causal abstractions of neural networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=RmuXDtjDhG.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. Finding alignments between interpretable causal variables and distributed neural representations. Ms., Stanford University, 2023. URL https://arxiv.org/abs/2303.02536.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability, 2024. URL https://arxiv.org/abs/2301.04709.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. *CoRR*, abs/2401.06102, 2024. doi: 10.48550/ARXIV.2401.06102. URL https://doi.org/10.48550/arXiv.2401.06102.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching, 2023.

Yaojie Hu and Jin Tian. Neuron dependency graphs: A causal abstraction of neural networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9020–9040. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/hu22b.html.

Jing Huang, Zhengxuan Wu, Christopher Potts, Mor Geva, and Atticus Geiger. Ravel: Evaluating interpretability methods on disentangling language model representations, 2024.

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4198–4205. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.386. URL https://doi.org/10.18653/v1/2020.acl-main.386.

Yibo Jiang, Goutham Rajendran, Pradeep Ravikumar, Bryon Aragam, and Victor Veitch. On the origins of linear representations in large language models. *CoRR*, abs/2403.03867, 2024. doi: 10.48550/ARXIV.2403.03867. URL https://doi.org/10.48550/arXiv.2403.03867.

Armin Kekic, Bernhard Schölkopf, and Michel Besserve. Targeted reduction of causal models. *CoRR*, abs/2311.18639, 2023. doi: 10.48550/ARXIV.2311.18639. URL https://doi.org/10.48550/arXiv.2311.18639.

Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *CoRR*, abs/2307.09458, 2023. doi: 10.48550/ARXIV.2307.09458. URL https://doi.org/10.48550/arXiv.2307.09458.

Zachary C. Lipton. The mythos of model interpretability. *Commun. ACM*, 61(10):36–43, 2018. doi: 10.1145/3233231. URL https://doi.org/10.1145/3233231.

Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. Towards faithful model explanation in NLP: A survey. *CoRR*, abs/2209.11326, 2022. doi: 10.48550/arXiv.2209.11326. URL https://doi.org/10.48550/arXiv.2209.11326.

Riccardo Massidda, Atticus Geiger, Thomas Icard, and Davide Bacciu. Causal abstraction with soft interventions. In Mihaela van der Schaar, Cheng Zhang, and Dominik Janzing, editors, *Conference on Causal Learning and Reasoning, CLeaR 2023, 11-14 April 2023, Amazon Development Center, Tübingen, Germany, April 11-14, 2023*, volume 213 of *Proceedings of Machine Learning Research*, pages 68–87. PMLR, 2023. URL https://proceedings.mlr.press/v213/massidda23a.html.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://aclanthology.org/N13-1090.

Aaron Mueller, Jannik Brinkmann, Millicent Li, Samuel Marks, Koyena Pal, Nikhil Prakash, Can Rager, Aruna Sankaranarayanan, Arnab Sen Sharma, Jiuding Sun, Eric Todd, David Bau, and Yonatan Belinkov. The quest for the right mediator: A history, survey, and theoretical grounding of causal interpretability, 2024. URL https://arxiv.org/abs/2408.01416.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Najoung Kim, Arya McCarthy, and Hosein Mohebbi, editors, *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2023, Singapore, December 7, 2023*, pages 16–30. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.BLACKBOXNLP-1.2. URL https://doi.org/10.18653/v1/2023.blackboxnlp-1.2.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *CoRR*, abs/2311.03658, 2023. doi: 10.48550/ARXIV.2311.03658. URL https://doi.org/10.48550/arXiv.2311.03658.

Eigil F. Rischel and Sebastian Weichwald. Compositional abstraction error and a category of causal models. In *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2021.

Paul K. Rubenstein, Sebastian Weichwald, Stephan Bongers, Joris M. Mooij, Dominik Janzing, Moritz Grosse-Wentrup, and Bernhard Schölkopf. Causal consistency of structural equation models. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Causal mediation analysis for interpreting neural nlp: The case of gender bias, 2020.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 11–20. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1002. URL https://doi.org/10.18653/v1/D19-1002.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=nRfClnMhVX.

Zhengxuan Wu, Atticus Geiger, Aryaman Arora, Jing Huang, Zheng Wang, Noah Goodman, Christopher Manning, and Christopher Potts. pyvene: A library for understanding and improving PyTorch models via interventions. In Kai-Wei Chang, Annie Lee, and Nazneen Rajani, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pages 158–165, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-demo.16. URL https://aclanthology.org/2024.naacl-demo.16.

Fabio Massimo Zennaro, Máté Drávucz, Geanina Apachitei, Widanalage Dhammika Widanage, and Theodoros Damoulas. Jointly learning consistent causal abstractions over multiple interventional distributions. In Mihaela van der Schaar and Cheng Zhang and/D Dominik Janzing, editors, *Conference on Causal Learning and Reasoning, CLeaR 2023, 11-14 April 2023, Amazon Development Center, Tübingen, Germany, April 11-14, 2023*, volume 213 of *Proceedings of Machine Learning Research*, pages 88–121. PMLR, 2023. URL https://proceedings.mlr.press/v213/zennaro23a.html.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Hf17y6u9BC.

Appendix A. Algorithms

There are two main algorithms our solution is based on: constructing evaluation graphs and partitioning inputs using these graphs. This section provides a detailed explanation of these approaches, including pseudocode and step-by-step examples.

A.1. Evaluation Graphs Explained

Each of the candidate causal models that we align with the LLM has a corresponding intervenable model trained to find these alignments, used to obtain its corresponding evaluation graph. The nodes in this graph correspond to the possible inputs of the task. For the sake of this example, we take only the case where in the arithmetic task, $X + Y + Z =$, we have $X, Y, Z \in \{1, 2\}$. Therefore, there are $2^3 = 8$ possible inputs. The possible inputs and their corresponding outcomes are:

- {'X': 1, 'Y': 1, 'Z': 1} with outcome 3.
- {'X': 1, 'Y': 1, 'Z': 2}, {'X': 1, 'Y': 2, 'Z': 1}, {'X': 2, 'Y': 1, 'Z': 1} with outcome 4
- {'X': 1, 'Y': 2, 'Z': 2}, {'X': 2, 'Y': 1, 'Z': 2}, {'X': 2, 'Y': 2, 'Z': 1} with outcome 5
- {'X': 2, 'Y': 2, 'Z': 2} with outcome 6

Therefore, we have 8 nodes in the evaluation graph. To construct the graph, we take each pair of nodes as input to a trained model for finding alignments between causal model $\mathcal{M}^k$ and a large language model. We call the trained model for finding alignments $I_{\mathcal{M}^k}$. We evaluate each input to be in turn the base and source for all the other nodes. For example, we evaluate both $I_{\mathcal{M}^k}(\texttt{base} = \{X : 1, Y : 1, Z : 1\}, \texttt{source} = \{X : 1, Y : 1, Z : 2\})$ and $I_{\mathcal{M}^k}(\texttt{base} = \{X : 1, Y : 1, Z : 2\}, \texttt{source} = \{X : 1, Y : 1, Z : 1\})$. Each evaluation yields a binary output (0 or 1), indicating the presence or absence of an alignment triggered by the counterfactual data obtained using that (base, source) unit. (Refer to Appendix C.2 for details on the counterfactual dataset used to train/evaluate $I_{\mathcal{M}^k}$.)

The edge between two nodes is determined as follows:

- No edge if $I_{\mathcal{M}^k}(\texttt{base}, \texttt{source}) = I_{\mathcal{M}^k}(\texttt{source}, \texttt{base}) = 0$, indicating no detected relationship in either direction.
- Edge weighted by 1 if $I_{\mathcal{M}^k}(\texttt{base}, \texttt{source}) = I_{\mathcal{M}^k}(\texttt{source}, \texttt{base}) = 1$, implying a strong alignment.
- Edge weighted by 0.5 if $I_{\mathcal{M}^k}(\texttt{base}, \texttt{source}) \neq I_{\mathcal{M}^k}(\texttt{source}, \texttt{base})$, suggesting a directional alignment.

Each graph has a corresponding IIA computed as:

$$\texttt{IIA}_{G^k} = 2 * \frac{\sum_{(i,j) \in E} w(i,j)}{|V| * (|V| - 1)} \tag{7}$$

where E is the set of edges in graph G^k, $|V|$ represents the number of nodes and $w(i, j)$ represents the weight between nodes i and j.

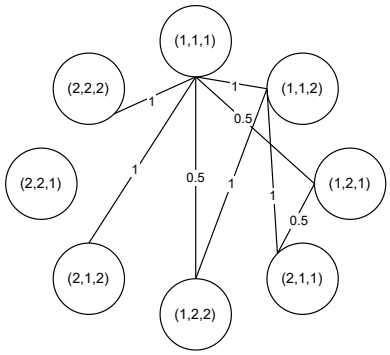


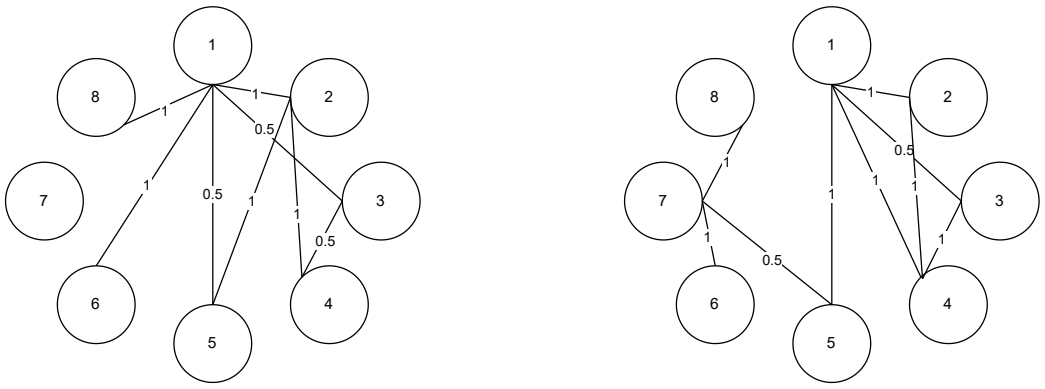
Figure 6: Example of an evaluation graph for solving the arithmetic task when $X, Y, Z \in \{1, 2\}$. There are 8 possible inputs, corresponding to the nodes in the graph. The lack of connections to node (2, 2, 1) indicates that a different, more appropriate intervenable model is likely required to understand how the LLM processes this particular input.

A fully connected graph with all edges weighted by 1 signifies a perfect alignment between the causal model $\mathcal{M}^k$ and the LLM's behaviour. Figure 6 shows an example of how the graph in the arithmetic task could look like when $X, Y, Z \in \{1, 2\}$. Algorithm 1 provides the pseudocode for constructing these evaluation graphs.

Algorithm 1: Construction of an evaluation graph for a candidate model.
Input: list of all possible inputs for a task V, trained alignment $I_{\mathcal{M}^k}$
Output: evaluation graph G^k
$G^k \leftarrow |V| \times |V|$ zero matrix
for $i \leftarrow 0$ ***to*** $|V| - 1$ **do**
  for $j \leftarrow i + 1$ ***to*** $|V| - 1$ **do**
    $base\_source \leftarrow$ counterfactuals when V[i] is the base and V[j] is the source
    $source\_base \leftarrow$ counterfactuals when V[i] is the source and V[j] is the base
    $G^k[i][j] = G^k[j][i] \leftarrow$ evaluate $I_{\mathcal{M}^k}$ on $[base\_source, source\_base]$
  end
end

A.2. Input Space Partitioning Explained

The core of our approach lies in partitioning the input space using the previously constructed evaluation graphs. This process aims to greedily maximize the portion of the input space explained within a defined faithfulness threshold, λ. In other words, the goal is to find a stronger hypothesis (combined model), where strength is defined by how little input space is left unassigned to an explanation.



(*a*) G^i (*b*) G^j

Figure 7: Example of two evaluation graphs, G^i and G^j, constructed by evaluating $I_{\mathcal{M}^i}$ and $I_{\mathcal{M}^j}$, respectively.

Figure 7 illustrates two evaluation graphs, G^i and G^j. We use them as examples to demonstrate how we combine their corresponding causal models to form $\mathcal{M}^{i+j}$ when the faithfulness level is $\lambda = 0.4$. Here is a step-by-step breakdown:

- We begin by choosing the graph with the highest IIA. Therefore, we start with G^j which has an IIA of $\frac{8}{28}$.
- The nodes of G^j are sorted in descending order based on their degree: 1,7,4,5,3,2,8,6.
- We greedily build a subgraph by iteratively adding nodes in the sorted order, ensuring the resulting subgraph maintains an IIA above the faithfulness threshold ($\lambda = 0.4$). This process results in the subgraph containing nodes 1, 7, 4, 5, and 2, with an IIA of 0.45.
- We move on to the next highest IIA graph, G^i, and order its nodes by degree in descending order: 1, 2, 5, 4, 3, 8, 6, 7.
- We exclude nodes already included in the subgraph from G_j, leaving the ordered list of nodes 3,8,6.
- There are no connections between the remaining nodes 3,8,6 in G^i. Therefore, adding these nodes would not enhance the explanation within the required faithfulness threshold.
- The portion of the input space that is left unassigned is handled by an output model. In this example, $\mathcal{M}^{i+j}$ explained 62% of the input space with 45% faithfulness.

This is just an example to showcase how partitioning the input space works. Algorithm 2 provides the pseudocode for implementing this input space partitioning based on evaluation graphs.

Algorithm 2: Greedy algorithm for selecting the inputs that correspond to each model.

Input: Faithfulness threshold λ, candidate models $\mathcal{M} = \{\mathcal{M}^1, ..., \mathcal{M}^k\}$, input space $\mathcal{X}$, evaluation graphs $\mathcal{G} = \{G^1, .., G^k\}$

Output: Set of selected models and their input partitions

assigned_inputs, selected_models $\leftarrow \emptyset, \emptyset$
while $\mathcal{X} \setminus$ *assigned_inputs* $\neq \emptyset$ **do**
  best_model $\leftarrow$ null
  best_inputs $\leftarrow \emptyset$
  foreach $\mathcal{M}^j \in \mathcal{M}$ **do**
    current_inputs $\leftarrow \emptyset$
    $g^k \leftarrow G^k(\mathcal{X} \setminus$ assigned_inputs$)$
    $temp\_g^k \leftarrow \emptyset$
    foreach $x \in$ *nodes sorted by degree in subgraph* g^k **do**
      $temp\_g^k \leftarrow temp\_g^k \cup \{x\}$
      if $IIA_{temp\_g^k} \geq \lambda$ **then**
        current_inputs $\leftarrow$ current_inputs $\cup \{x\}$
      end
    end
    if $|$*current_inputs*$| > |$*best_inputs*$|$ **then**
      best_model $\leftarrow \mathcal{M}^j$
      best_inputs $\leftarrow$ current_inputs
    end
  end
  if *best_model* = *null* **then**
    Break
  end
  selected_models $\leftarrow$ selected_models $\cup \{$best_model$\}$
  assigned_inputs $\leftarrow$ assigned_inputs $\cup$ best_inputs
end
return (selected_models, assigned_inputs)

Appendix B. LLM Selection and Fine-tuning

The fine-tuning process was implemented using the Hugging Face Transformers library, which offers a comprehensive toolkit for working with pre-trained language models. Because for our arithmetic or boolean logic task the goal is to predict the correct output of the sum of three numbers or the correct evaluation of a logic expression, respectively, the language modelling capabilities of GPT-2 are not needed. Therefore, we employ the `GPT2ForSequenceClassification` class, a specialized variant of GPT-2 designed for sequence classification tasks. The fine-tuning process involved a selection of hyperparameters to optimize performance on the two tasks. The key hyperparameters and their chosen values are presented in Table 1.

Hyperparameter	**Value**
Training Size	2560
Learning Rate	2e-5
Optimizer	AdamW
Batch Size	32
Epochs	50

Table 1: Hyperparameters for fine-tuning the GPT2 to the arithmetic and boolean logic task.

Appendix C. Data Generation

The data which is needed throughout the pipeline of this research is of two types: factual, which is observational data used for fine-tuning the LLM, and counterfactual, which is interventional data used to find alignments. Utilizing black box models, such as LLMs, has the advantage of easily generating counterfactual data. Black boxes are closed systems, where the internal mechanisms are abstracted away. By simply manipulating the input parameters and observing the corresponding outputs, one can explore different hypothetical scenarios, effectively simulating counterfactual conditions. Since they are closed systems, the output can be simultaneously analysed in factual and counterfactual contexts.

C.1. Factual Datasets

To obtain a *factual dataset*, one must construct a standard input-taking prompt. The inputs must match the input variables in the causal models that one wishes to align. All the causal models which solve the same task share the same factual dataset. For the arithmetic task that we employ, we generate factual data using the prompt below, where the inputs X, Y, Z are numerical values between 1 and 10:

```
X+Y+Z=
```

For the arithmetic task, we finetune the GPT2-small generating prompts of the form, where the inputs are defined as in the main paper:

OP1 (OP_2 (X) B OP_3 (Y)) =

C.2. Counterfactual Datasets

Generating the counterfactual data requires two generated inputs: a base and a source. Depending on the intervenable variable, the counterfactual output is obtained by replacing the base input under that intervenable variable with the source sample under the same intervenable input.

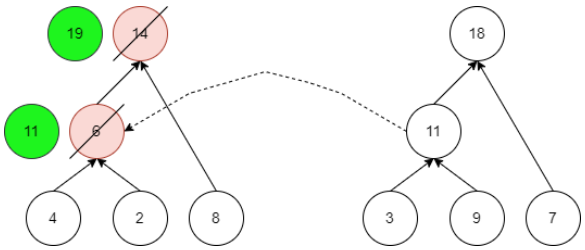


Figure 8: Example of obtaining counterfactual data where the base is $\{X : 4, Y : 2, Z : 8\}$ with intermediate variable $P = X + Y = 6$ and output $O = P + Z = 14$, and the source is $\{X : 3, Y : 9, Z : 7\}$ with intermediate variable $P = X + Y = 11$ and output $O = P + Z = 18$.

For example, in Figure 8, one can visualize how such counterfactual data is obtained for the arithmetic task. The first level in the trees represents the input, everything that is in between the first level and the last represents the intervenable variables, and the last level represents the outcome. In the example graphs, the base is the tuple (4, 2, 8) with outcome 14, and the source is the tuple (3, 9, 7). Applying an *interchange intervention (II)* between the source and the base at the intervenable variable means replacing the outcome of the intervenable variable of the base, which is 6, with the outcome of the intervenable variable of the source, which is 11. Therefore, the counterfactual outcome becomes 19.

C.3. Generation of Graph Nodes

One essential part of our analysis is to construct the graph $G = (V, E, w)$. In this section, we explain how the set of vertices, V, is obtained. The input to the arithmetic task is constrained to the integer interval [1, 10]. Each node within the graph is represented by a tuple of three such integers. We systematically generate all possible permutations of three numbers within this interval, associating each tuple with a distinct node. Given that the number of arrangements of three elements from a set of ten is 10 * 10 * 10 = 1000, the graph consists of 1000 nodes. The set of nodes V in graph G is formally defined in Equation 8.

$$V := \{(X, Y, Z) \mid X, Y, Z \in 1, 2, ..., 10\} \tag{8}$$

Similarly, for the boolean logic task, we have two possible values for each of the 6 inputs OP_1, OP_2, X, B, OP_3, Y, and every combination of their values results in $2 * 2 * 2 * 2 * 2 * 2 = 64$ possible inputs.

Appendix D. Training Intervenable Models

Given the structure of our arithmetic task, interventions are performed on up to 6 tokens simultaneously, corresponding to the maximum number of tokens in the prompt $'X + Y + Z ='$. Besides the explored subspace dimensions, $low\_rank\_dimension \in \{64, 128, 256\}$, for causal models M_X and M_O we also test for lower dimensions such as $low\_rank\_dimension \in \{4, 8, 16, 32\}$. By varying the subspace dimension, we can assess the trade-off between computational efficiency and the ability to capture nuanced representations of the arithmetic operations within these lower-dimensional spaces. In total, there are 216 trained intervenable models for each of the 6 causal models in the arithmetic task, because on each of the 12 layers, for each of the three low-rank dimensions $\{64, 128, 256\}$, an alignment is searched for through training (216 trained intervenable models), and for two of the causal models, an additional of 96 models are trained, because we also search alignments in 4 additional low rank dimensions $\{4, 8, 16, 32\}$. For each of the tasks, the training hyperparameters are shown in Table 2.

Hyperparameter	**Arithmetic**	**Boolean**
Training Size	256000	4096
Learning Rate	0.01	0.01
Batch Size	1280	128
Epochs	4	5

Table 2: Hyperparameters for training intervenable models for the arithmetic and boolean tasks.

Appendix E. Models For The Boolean Task

The models below represent the intermediate states GPT2-small could be in when solving the boolean logic task. The explicit function definitions are listed below.

$$\begin{aligned}
\mathcal{M}^{X}&: \ \mathcal{F}_P(X) = X, & \mathcal{F}_O(OP_1, OP_2, P, B, OP_3, Y) &= OP_1\,(OP_2(P)\,B\,OP_3(Y))\\
\mathcal{M}^{Y}&: \ \mathcal{F}_P(Y) = Y, & \mathcal{F}_O(OP_1, OP_2, X, B, OP_3, P) &= OP_1\,(OP_2(X)\,B\,OP_3(P))\\
\mathcal{M}^{B}&: \ \mathcal{F}_P(B) = B, & \mathcal{F}_O(OP_1, OP_2, X, P, OP_3, Y) &= OP_1\,(OP_2(X)\,P\,OP_3(Y))\\
\mathcal{M}^{OP_1}&: \ \mathcal{F}_P(OP_1) = OP_1, & \mathcal{F}_O(P, OP_2, X, B, OP_3, Y) &= P\,(OP_2(X)\,B\,OP_3(Y))\\
\mathcal{M}^{OP_2}&: \ \mathcal{F}_P(OP_2) = OP_2, & \mathcal{F}_O(OP_1, P, X, B, OP_3, Y) &= OP_1\,(P(X)\,B\,OP_3(Y))\\
\mathcal{M}^{OP_3}&: \ \mathcal{F}_P(OP_3) = OP_3, & \mathcal{F}_O(OP_1, OP_2, X, B, P, Y) &= OP_1\,(OP_2(X)\,B\,P(Y))
\end{aligned}$$

$$\begin{aligned}
\mathcal{M}^{X'}&: \ \mathcal{F}_P(OP_2, X) = OP_2(X), & \mathcal{F}_O(OP_1, P, B, OP_3, Y) &= OP_1\,(P\,B\,OP_3(Y))\\
\mathcal{M}^{Y'}&: \ \mathcal{F}_P(OP_3, Y) = OP_3(Y), & \mathcal{F}_O(OP_1, OP_2, X, B, P) &= OP_1\,(OP_2(X)\,B\,P)\\
\mathcal{M}^{Q}&: \ \mathcal{F}_P(OP_2, X, B, OP_3, Y) = OP_2(X)\,B\,OP_3(Y), & \mathcal{F}_O(OP_1, P) &= OP_1\,(P)\\
\mathcal{M}^{V}&: \ \mathcal{F}_P(OP_1, OP_2, X) = OP_1(OP_2(X)), & \mathcal{F}_O(OP_1, P, B, OP_3, Y) &= P\,OP_1(B)\,OP_1(OP_3(Y))\\
\mathcal{M}^{W}&: \ \mathcal{F}_P(OP_1, OP_3, Y) = OP_1(OP_3(Y)), & \mathcal{F}_O(OP_1, OP_2, X, B, P) &= OP_1(OP_2(X))\,OP_1(B)\,P\\
\mathcal{M}^{B'}&: \ \mathcal{F}_P(OP_1, B) = OP_1(B), & \mathcal{F}_O(OP_1, OP_2, X, P, OP_3, Y) &= OP_1(OP_2(X))\,P\,OP_1(OP_3(Y))
\end{aligned}$$

$$\mathcal{M}^{O}: \ \mathcal{F}_P(OP_1, OP_2, X, B, OP_3, Y) = OP_1\,(OP_2(X)\,B\,OP_3(Y)), \qquad \mathcal{F}_O(P) = P$$