Causal Path Tracing in Transformers

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose a causal path tracing framework to understand how information causally flows through the internal structures of transformers for a given decision. By unfolding each block into a causal graph of path nodes and applying a *minimality-based subset search*, our method identifies all possible causal paths within each block, with polynomial-time complexity on average. Furthermore, we demonstrate the reliability of a *union-based causal path reference strategy*, enabling efficient and reliable causal tracing throughout the model. The key contributions of this work are: (1) an automated, efficient framework for causal path tracing that exhaustively searches paths along direct dependencies; (2) theoretical and empirical validation demonstrating exhaustive search with polynomial-time complexity on average; (3) experimental findings showing that self-repair effects occur far less frequently along the identified causal paths, that certain paths are uniquely activated for specific classes, and that the traced paths are both accurate and faithful.

1 Introduction

With the success of transformers [1] across language and vision, interest has grown in understanding their internal mechanisms beyond their black-box nature, especially to enable safer deployment in high-stakes applications such as healthcare, law, and education. Mechanistic interpretability aims to identify specific components within the model, such as attention heads or MLPs, that contribute to its behavior. Building with mathematically grounded circuit discovery in simplified settings [2], recent efforts have incorporated Pearl's causal theory [3], employing ablation-based interventions to trace which parts of the network support particular outputs.

Depending on the granularity of analysis, prior work can be classified into: node-level patching [4, 5, 6, 7, 8, 9], which identifies the role of individual input features; edge-level patching [10, 11, 12], which examines the influence of neighboring feature pairs with direct computational dependencies; and path-level patching [13, 14, 15], which investigates the contribution of distant feature pairs connected through multiple accumulated dependencies.

Recent work [16] has shown that ablation-based methods often fail to estimate true causal effects due to self-repair (or backup behavior), where later components compensate for earlier ablations. This implies that, when unablated components lie between the target and the decision, internal explanations may be misattributed. To address this, one solution is to iteratively evaluate each component conditioned on priorly identified causal components along direct computational dependencies; here, we refer to as *causal referencing*. However, prior node- or edge-level approaches cannot fully support causal referencing over all relevant combinations; though sequentially feasible, it remains inaccurate. In contrast, path-level approaches can in principle support this, but due to their combinatorial complexity, existing studies typically rely on hypothetical tests [13] or assess a single subpath manually [14, 15], making it infeasible to capture a full explanation for a decision in time. (Table 1)

To address this obstacle, we propose an automated and efficient framework for tracing causal paths given a decision. Specifically, we begin by unfolding all possible paths within each block of a

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Approach	Patching	Path Tracing for Decision	
		Feasibility	**Reliability**
[4, 5, 6, 7, 8, 9]	Node	✓ backward chaining only	✗ no causal referencing
[10, 11, 12]	Edge	✓ backward chaining only	✗ no causal referencing
[13]	Path	✗ hypothetical only	✓ full coverage
[14, 15]	Path	✗ manual subpath	✓ full coverage
Ours	Path	✓ polynomial on average	✓ full coverage

Table 1: **Comparison of patching methods for path tracing in a given decision.** Feasibility refers to empirical applicability for a given decision; reliability to its theoretical guarantees.

transformer, interpreted as a causal graph, into path nodes. Then, by introducing a minimality-based subset search strategy for identifying all possible causal path node combinations per block, we reduce the inherently exponential complexity to polynomial time on average. Furthermore, to enable efficient block-wise tracing, we demonstrate that referencing the union of causal paths identified in preceding blocks not only makes this feasible but also ensures reliability.

Our approach reveals that self-repair occurs primarily outside the identified causal path; thus, the path contains information essential to the decision and not easily replaced, reflecting its critical role. Moreover, we found that there exist causal paths uniquely associated with specific classes. These paths are activated only for their corresponding classes, serving class-specific roles within the model. Taken together, our results show that the proposed method faithfully and accurately explains model behavior under empirical evaluation.

2 Methodology

2.1 Preliminaries

To proceed, we introduce the definitions used throughout this work. Our goal is to reveal and explain internal components for decision by efficiently tracing causal paths. To enable this, following Pearl's causal theory [3], we interpret the transformer as a causal graph, as formalized in Definition 1. Based on this interpretation, we define the causal path for a given model decision through Definitions 2 to 4, where Definition 4 is adapted from the Halpern–Pearl definition of actual causality [17, 18].

Definition 1 (Transformer as Causal Graph)**.** *We say that a transformer is a **causal graph** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v \in \mathcal{V}$ denotes an internal component (e.g., intermediate feature) and each edge $v_i \to v_j \in \mathcal{E}$ indicates a direct computational dependency, if it satisfies the following conditions:*

(1.a) ***Directed Acyclic Graph:*** *Its internal computation proceeds layer by layer in a forward direction without cycles, which naturally forms a directed acyclic graph structure.*

(1.b) ***Markov:*** *Each node is deterministically computed from its parent nodes. This ensures that each node is conditionally independent of its non-descendants, given its parents.*

(1.c) ***Causal Sufficiency:*** *All nodes involved in its internal computation are observable, with no latent confounders or hidden common causes among nodes.*

Definition 2 (Model Decision)**.** *Let $y \in \mathbb{R}^{\mathcal{C}}$ be the model's output over $\mathcal{C}$ classes. We define the **model decision** as the index c^* such that $y^{(c^*)} > y^{(i)}$ for all $i \neq c^*$, i.e., the strict argmax.*

Definition 3 (Causal Path)**.** *Given a transformer interpreted as a causal graph $\mathcal{G}$, we define a **causal path** as a sequence of causal node sets $\mathcal{P} = (V_1, V_2, \dots)$, where each $V_i \subseteq \mathcal{V}$ is a **causal node set**, and every node $v \in V_i$ is connected via a directed edge to either the model input, the model output y, or a node in another causal node set V_j with $j \neq i$.*

Definition 4 (Causal Node Set)**.** *Given a transformer interpreted as a causal graph $\mathcal{G}$, a causal subpath reference $P \subseteq \mathcal{P}$ connecting a node set $V \subseteq \mathcal{V}$ to the output, and an off-path node set $\hat{V}$ such that $P \cup \hat{V}$ equals the set of all nodes between V and the output, and $P \cap \hat{V} = \emptyset$, we say that V is **causal** for the decision if the following conditions are satisfied:*

(4.a) ***Necessity (Counterfactual):*** *Let V' denote that V is intervened on to take a different value, and let $\hat{V}'$ be defined analogously for $\hat{V}$ to causally isolate V. Under these interventions, the output y' satisfies $\arg\max_i y'^{(i)} \neq c^*$, where c^* denotes the decision without any intervention.*

(4.b) ***Sufficiency (Contingency):*** *Given V, even if nodes in P are perturbed due to an intervention resulting in $\hat{V}'$, the output y' satisfies $\arg\max_i y'^{(i)} = c^*$.*

(4.c) ***Causal Minimality:*** *V is minimal; no strict subset of V satisfies both (4.a) and (4.b).*

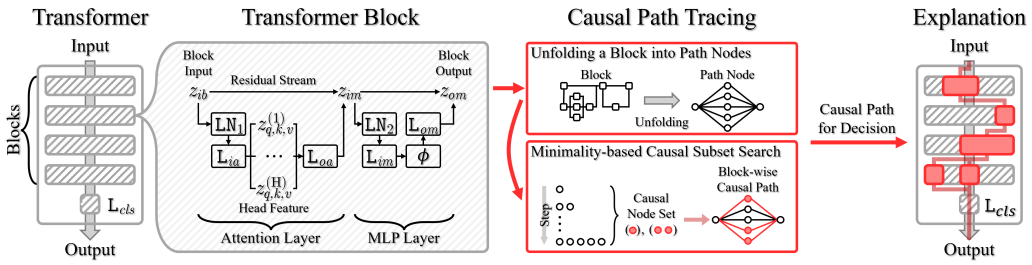


Figure 1: **Overview of our causal path tracing.** L: linear layer, LN: layer normalization, z: feature.

Having established the definitions with respect to causal paths, we present the structural Property 1 that highlights their recursive nature: under a given decision, any causal node set must have at least one parent node set that is also causal. This recursive property enables us to identify exhaustive causal paths by ensuring that causal influence can be traced backward through successive parent node sets.

Property 1 (Causal Edge: Existence of Causal Parent for Any Causal Node)**.** *In a transformer interpreted as a causal graph $\mathcal{G}$, each child node has a direct computational dependency (i.e., edges) with its parent nodes. Thus, under sufficient intervention, for every child node in $\mathcal{G}$ that belongs to a causal node set (possibly not minimal), at least one of its parent node sets is also a causal node set (possibly not minimal), given the decision. Here, "possibly not minimal" indicates that the node set may satisfy only the intervention-based conditions (Conditions (4.a) and (4.b)) without the minimality condition (Condition (4.c)). Although such a set may not strictly qualify as a causal node set under the full definition, the minimality condition is dependent on the other two and is not required for this property to hold.*

Example 1. *Consider a transformer interpreted as $\mathcal{G}$, containing only a node v_c and the set V_p of all parent nodes of v_c. Suppose that v_c belongs to a causal node set (i.e., the causal node set consists of v_c alone). Since v_c is deterministically computed from V_p, intervening on all of V_p leads to a different decision, satisfying Condition (4.a). Moreover, if V_p is unchanged, the decision remains the same since there is no off-path node set (i.e., $\hat{V} = \emptyset$), thereby satisfying Condition (4.b). Thus, V_p satisfies both conditions in Definition 4, implying that it is, at least, a causal node set.*

2.2 Intervention for Causal Isolation

As established in the preceding definitions, applying an intervention to isolate a node set is essential for identifying its causal influence in transformers. While there are various possible forms of intervention, it is not always clear whether they guarantee causal isolation under our setting. To address this, we formally define a *sufficient intervention* in Definition 5, to serve as a basis for assessing whether an intervention achieves reliable causal isolation in our framework.

Definition 5 (Sufficient Intervention)**.** *Given a transformer interpreted as a causal graph $\mathcal{G}$, we say that a node set V is **sufficiently intervened** to V' if the following conditions are satisfied:*

*(5.a) **Causal Structural Isomorphism:** The graphical structure of V in $\mathcal{G}$, namely the adjacency structure between V and its neighboring nodes, must differ from that of V', and their corresponding mathematical structures (i.e., structural equations) must likewise differ. This reflects a one-to-one correspondence between graphical and mathematical structures.*

*(5.b) **Causal Edge Validity:** Given that Property 1 holds in $\mathcal{G}$, it must still hold even after an intervention on V. That is, the intervention must not violate the conditions specified in Definitions 1 to 4 with respect to causal paths in $\mathcal{G}$.*

*(5.c) **Intervention Controllability:** An intervention must not be parametrically non-controllable; its effect must remain interpretable, rather than being overwhelmed by the parameters of the intervention, such as stochastic variations within them.*

Among possible intervention strategies within transformers, a straightforward approach is to add noise directly to the target node (DIRECT NOISE). However, such perturbation fails to satisfy Condition (5.a). Another commonly used strategy, as in prior works such as [6], involves forwarding a noise-perturbed token embedding through the model to obtain a corrupted version of the target node (NOISE TOKEN);

however, this violates Condition (5.c). A naive alternative, such as zero-masking (ZERO MASK), also proves inadequate, as it breaks Condition (5.b) by distorting Property 1.

Since these strategies fail to satisfy the required conditions in our setting, we instead adopt a method that intervenes on the target node by resampling from alternative token embeddings (TOKEN RESAMPLING). This approach satisfies all three conditions for a sufficient intervention, as demonstrated by the example in Appendix.

2.3 Unfolding Transformer Block

In this section, we introduce a mathematical formulation of paths within a standard transformer. We begin by representing paths in a single block to establish the idea of our approach. Subsequently, we extend this formulation to cover all possible paths from the given input to the output.

Given an input x, our goal is to identify *which structures* within the transformer contribute to the decision as causal paths. This requires identifying the causal node sets from the decision, which in turn involves exploring the model in the backward direction. To this end, we first decompose a single block into circuits, i.e., paths, as follows (notation is provided in Figure 1):

$$\begin{aligned}
\left[\, [z_q^{(h)}]_{h=1}^{H};\ [z_k^{(h)}]_{h=1}^{H};\ [z_v^{(h)}]_{h=1}^{H} \,\right] &= \mathtt{L}_{ia}(\mathtt{LN}_1(z_{ib})),\\
z_{oa} &= \mathtt{L}_{oa}([\,\mathrm{softmax}(z_q^{(h)} z_k^{(h)\top}/\sqrt{d_h}) z_v^{(h)}\,]_{h=1}^{H}),\\
z_{im} &= z_{ib} + z_{oa},\\
z_{om} &= \mathtt{L}_{om}\left(\phi(\mathtt{L}_{im}(\mathtt{LN}_2(z_{im})))\right),\\
z_{ob} &= z_{im} + z_{om},
\end{aligned} \tag{1}$$

where $z_{ib}, z_{oa}, z_{im}, z_{om}, z_{ob} \in \mathbb{R}^{T\times d_m}$ and $z_q^{(h)}, z_k^{(h)}, z_v^{(h)} \in \mathbb{R}^{T\times d_h}$, with T denoting the number of tokens, d_m the model dimension, and $d_h = {}^{d_m}\!/_{H}$. Here, to identify the structures causally involved in the decision from the input, we treat the block input z_{ib} as a single node. If the block output z_{ob} can be unfolded with respect to z_{ib}, this allows us to capture the structures in a path-wise manner ($z_{ib} \sim z_{ob}$), all at once, rather than laboriously analyzing them one by one in a structure-wise fashion.

However, due to the presence of non-linear functions, i.e., softmax and GeLU ϕ, it is nontrivial to decompose the above equations into a single unified expression. To address this, we employ a minor computational trick that rewrites the non-linear functions in the form of Hadamard products: $\mathrm{softmax}(z/\sqrt{d_h}) = z \odot D_\alpha$ and $\phi(z) = z \odot D_\beta$, where the scaling factors D_α and D_β are treated as fixed values once computed from the input z. In addition, we apply a similar simplification to layer normalization by treating its input-dependent statistics, mean and variance, as fixed after computation: $\mathtt{LN}(z) = zW_{ln}^\top + b_{ln}$. Together, these interpretations allow us to express the block in a form that structurally resembles a composition of linear operations and element-wise products, as follows:

$$\begin{aligned}
z_{oa} &= (\sum_{h=1}^{H} z_q^{(h)} z_k^{(h)\top} \odot D_\alpha z_v^{(h)} W_{oa}^\top) + b_{oa},\\
z_{om} &= z_{oa} W_{ln_2}^\top W_{im}^\top \odot D_\beta W_{om}^\top + z_{ib} W_{ln_2}^\top W_{im}^\top D_\beta W_{om}^\top\\
&\quad + b_{ln_2} W_{im}^\top \odot D_\beta W_{om}^\top + b_{im} \odot D_\beta W_{om}^\top + b_{om},\\
z_{ob} &= z_{ib} + z_{oa} + z_{om}\\
&= \underbrace{z_{ib}}_{\text{Residual Only (1 Path)}} + \underbrace{\sum_{h=1}^{H} z_q^{(h)} z_k^{(h)\top} \odot D_\alpha z_v^{(h)} W_{oa}^\top + b_{\text{attn}}}_{\text{Attention Only (H Paths)}} + \underbrace{z_{ib} W_{ln_2}^\top W_{im}^\top D_\beta W_{om}^\top + b_{\text{mlp}}}_{\text{MLP Only (1 Path)}}\\
&\quad + \underbrace{\sum_{h=1}^{H} z_q^{(h)} z_k^{(h)\top} \odot D_\alpha z_v^{(h)} W_{oa}^\top W_{ln_2}^\top W_{im}^\top \odot D_\beta W_{om}^\top + b_{\text{attn+mlp}}}_{\text{Attention+MLP (H Paths)}},
\end{aligned} \tag{2}$$

Here, $b_{\text{attn}}, b_{\text{mlp}}$, and $b_{\text{attn+mlp}}$ represent the terms in the block output z_{ob} that do not directly involve the block input z_{ib} (the full derivation is provided in Appendix). By unfolding z_{ob} with respect to z_{ib}, we obtain a set of additive terms, which can be grouped into distinct paths depending on whether they contain z_{ib} as a multiplicative factor. Ultimately, we can *treat each of these paths as a single node* to assess its causal contribution.

Algorithm 1 Minimality-based Causal Subset Search per Block

Input: A path node set $V_p = [v_1, \ldots, v_n]$ from a specific block (i.e., additive terms within the block), a subgraph $\mathcal{G}_c$ (downstream blocks of V_p in the transformer $\mathcal{G}$), causal subpaths P connecting V_p to the decision c^* in the output y, and an off-path node set $\hat{V}$ such that $P \cup \hat{V}$ equals all nodes in $\mathcal{G}_c$ and $P \cap \hat{V} = \emptyset$
Output: $V_{\text{out}} = \{V_1, V_2, \ldots\}$, where each $V_i \subseteq V_p$ satisfies Conditions (4.a), (4.b), and (4.c)
$V_{\text{out}} \leftarrow \emptyset$
for $s = 1$ to n **do** ▷ Subset size
  for each $V \subseteq V_p$ such that $|V| = s$ **do**
    if $V_i \subseteq V$ for some $V_i \in V_{\text{out}}$ **then**
      continue ▷ Fail Condition (4.c) (causal minimality)
    end if
    Intervene on $V \to V'$, and on $\hat{V} \to \hat{V}'$
    Let $y' \leftarrow$ model output under $(V', \hat{V}', P)$ ▷ for Condition (4.a) (necessity)
    Let $y'' \leftarrow$ model output under $(V, \hat{V}', P)$ ▷ for Condition (4.b) (sufficiency)
    if $\arg\max_i y'^{(i)} = c^*$ **or** $\arg\max_i y''^{(i)} \neq c^*$ **then**
      continue ▷ Fail Condition (4.a) or (4.b)
    end if
    $V_{\text{out}} \leftarrow V_{\text{out}} \cup \{V\}$ ▷ Satisfies Conditions (4.a), (4.b), (4.c)
  end for
end for
return V_{out}

Note that we omit the unfolding of $z_q^{(h)}$, $z_k^{(h)}$, and $z_v^{(h)}$ for brevity, as they are linear functions of z_{ib} via W_{ln_1}, W_{ia} and follow the same path structure. Furthermore, we assume that bias terms explicitly excluding z_{ib} propagate uniformly their influence across all paths through their originating layers.

2.4 Minimality-based Causal Subset Search per Block

As shown earlier, all paths within a block can be decomposed into additive terms, each treated as an individual node. Based on this, we perform a block-wise backward search for causal node sets to trace the causal path for a given decision. Here, since path-level interactions must be considered, all possible combinations of path nodes within each block need to be evaluated. However, a brute-force approach incurs a complexity of $O(2^n)$, as this subset search problem is NP-complete, making it impractical for large-scale search. Although NP-complete problems cannot be solved in polynomial time in the worst case, we propose a strategy based on Condition (4.c) that enables polynomial-time search on average.

The core idea, based on Condition (4.c), is that a causal node set must be minimal. That is, if a subset $V \subseteq V_p$ is identified as a causal node set, where V_p denotes the set of all path nodes within a block, then any superset of V cannot be minimal and thus does not need to be evaluated. Building on this, our search strategy proceeds in steps by subset size, starting from the smallest. As illustrated in Algorithm 1, causal node sets identified at smaller steps are used to prune the search space at larger steps by eliminating supersets that violate minimality. This strategy leads to an average-case time complexity that is polynomial in practice, as formally analyzed in Theorem 1 (proof in Appendix).

Theorem 1 (Expected Time Complexity of Minimality-based Subset Search)**.** *Consider a minimality-based subset search over n nodes, where each subset is independently selected as a causal node set with probability p. Then, the expected number of subset evaluations over all subsets is bounded by:*

$$n + (1-p) \times \sum_{s=2}^{n} \max\left(0, \binom{n}{s} + \sum_{i=1}^{s-1} \sum_{m=1}^{\lfloor p\binom{n}{i} \rfloor} (-1)^m \binom{p\binom{n}{i}}{m} \binom{n-mi}{s-mi}\right). \tag{3}$$

Given this, the expected time complexity grows approximately as:

$$O\left(n^{\lfloor \log_2\left(\frac{1}{p}+2\right) \rfloor}\right). \tag{4}$$

Remark 1. *Although the exact value of p is unknown, the time complexity, depending on p, is polynomial in the best and average cases. For example, when $p = 1$, all subsets of size $s \geq 2$ are pruned, so only singleton subsets are evaluated, resulting in a time complexity of $O(n)$. However,*

Algorithm 2 Unfolded Block-wise Causal Path Tracing

Input: A transformer $\mathcal{G}$ with D blocks, and a model output y with decision c^* for a given input
Output: Causal paths $\mathcal{P} = (V_{\text{out}}^{(D)}, \ldots, V_{\text{out}}^{(1)})$, a sequence of causal node sets identified per block
$P \leftarrow \{\mathtt{L}_{cls}\}$ ▷ By Property 1, the classifier $\mathtt{L}_{cls}$ serves as the initial causal path reference
$\mathcal{P} \leftarrow \emptyset$; $\mathcal{G}_c \leftarrow \{\mathtt{L}_{cls}\}$
for $j = D$ to 1 **do** ▷ Iterate backward through transformer blocks
    Let $V_p^{(j)} \leftarrow$ unfolded path nodes in block j
    $V_{\text{out}}^{(j)} \leftarrow \text{MIN\_SEARCH}(V_p^{(j)}, \mathcal{G}_c, P, c^*)$ ▷ See Algorithm 1
    $P \leftarrow \bigcup V_{\text{out}}^{(j)}$ ▷ Update causal path reference (see Theorem 2)
    $\mathcal{P} \leftarrow \mathcal{P} \cup V_{\text{out}}^{(j)}$; $\mathcal{G}_c \leftarrow$ block j
end for
return $\mathcal{P}$

since the problem is fundamentally NP-complete, exponential complexity is unavoidable in the worst case. Nonetheless, such worst-case scenarios occur only infrequently; for example, when $p \leq \frac{1}{2^n-2}$, causal node sets are rarely selected at each step, requiring exhaustive search over all subset combinations and leading to a time complexity of $O(2^n)$.

2.5 Unfolded Block-wise Causal Path Tracing

In this section, we extend the minimality-based causal subset search from a single block to the entire transformer. We traverse blocks backward, identifying causal node sets and updating the causal path reference P at each step. Using each causal set individually as P is computationally expensive, as it requires repeated searches. Instead, we use their union as the reference, which significantly reduces the cost. As in Theorem 2, the union-based strategy ensures that reliability converges to 1 (proof in Appendix), indicating near-complete causal coverage. Algorithm 2 outlines the full procedure.

Theorem 2 (Causal Union Reference Reliability)**.** *Consider a minimality-based subset search over n nodes, where each subset is independently selected as a causal node set with probability p. Suppose that a collection of such sets, $V_{out}^{(j+1)} = \{V_i^{(j+1)}\}_{i=1}^k$, is identified from the $(j+1)$-th block, i.e., the one directly downstream. Their union, denoted as $P = \bigcup_{i=1}^k V_i^{(j+1)}$, serves as the causal subpath reference for the minimality-based subset search in the j-th block. Let s_{avg} denote the average size of the k causal node sets in $V_{out}^{(j+1)}$. Then, the reliability of the resulting causal node set obtained using P is given by:*

$$p + (1-p)\left(1 - \left(1 - \frac{s_{avg}}{n}\right)^k\right)^n \to 1 \tag{5}$$

3 Experiments

3.1 Models, Datasets, and Baselines

We conduct experiments on five transformer models: three language models (GPT2-xs [19], Pythia-14m and Pythia-1b [20]) and two vision models (ViT-tiny [21] and DeiT-tiny [22]). For language tasks, we use the KNOWNS1000 [6] and T-REX [23, 24] datasets. For vision tasks, we evaluate on IMAGENET [25] and OFFICEHOME [26]. Due to space constraints, further results are provided in Appendix.

As summarized in Table 1, we compare against existing methods that are feasible for decision-level path tracing. To enable fair comparison with our method, all baselines are extended under a backward chaining framework, assuming that residual connections are always present—even when reliability conditions are not met.

Specifically, NT_1 and $\text{NT}_{10\%}$ are adaptations of the node-level patching method from [6], referred to as Node-level patching-based Tracing (NT), where the top-1 node (NT_1) or the top 10% of nodes ($\text{NT}_{10\%}$), ranked by their estimated effect within each block, are selected as decision paths. ET_{all} and ET_{cls} are based on the edge-level patching method from [12], referred to as Edge-level

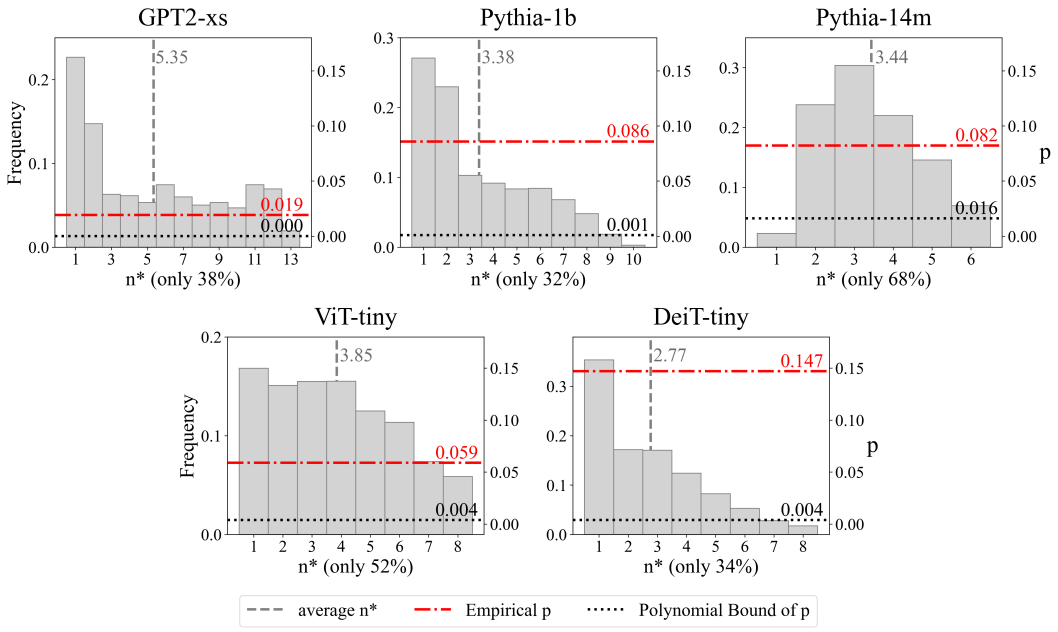


Figure 2: **Comparison of empirical time complexity.** Causal path tracing under our method runs in polynomial time across models. Each subplot shows the reduced search space (in parentheses); n^* is the maximum step reached by the minimality-based search (i.e., the largest s such that the term in Equation (3) is nonzero); the empirical p estimated from the average n^* (see Theorem 1); and the polynomial bound of p, which is the theoretical lower bound required to ensure polynomial-time search (see Remark 1). Language models use T-Rex; vision models use ImageNet.

patching-based Tracing (ET), which assumes task-level edge attribution. Here, a "task" is defined as either the entire dataset (ET_{all}) or a single class (ET_{cls}).

Note that path-level patching methods are not included in the comparison, as no existing method feasibly enumerates all decision paths for a given output—our method is the first to make this feasible. We refer to our approach as Causal Path Tracing (CPT). Implementation details are in Appendix.

3.2 Results

Minimality-based search converges empirically in polynomial time; furthermore, it reveals how models rely on path-level reasoning Figure 2 presents the empirical time complexity of our causal path tracing procedure across models. Each subplot shows the distribution of n^*, defined as the final step in the minimality-based search where no further superset remains due to pruning by already selected causal node sets. The average n^* is used to estimate the empirical probability p that a randomly selected subset is causal (see Theorem 1), which is then compared against the theoretical lower bound required for polynomial-time search (see Remark 1).

In all models, the empirical p exceeds the theoretical threshold, confirming that the proposed search converges in polynomial time in practice, as predicted. Notably, the search typically completes in few steps, with pruning often concluding well before the midpoint of the search space.

The distribution of n^* also reveals how the model leverages internal structure for decision making: a small n^*, especially when concentrated near one, indicates that the model relies primarily on the strength of individual paths; in contrast, a larger or more dispersed n^* suggests that reasoning involves interactions among multiple paths rather than relying on any single strong one.

Causal path components exhibit lower self-repair, suggesting irreplaceable decision signals We compare self-repair scores between attention heads on the causal path and those off the path, as identified by our tracing method. Following the prior work [16], we categorize components based on whether they belong to the traced causal path and measure their self-repair accordingly.

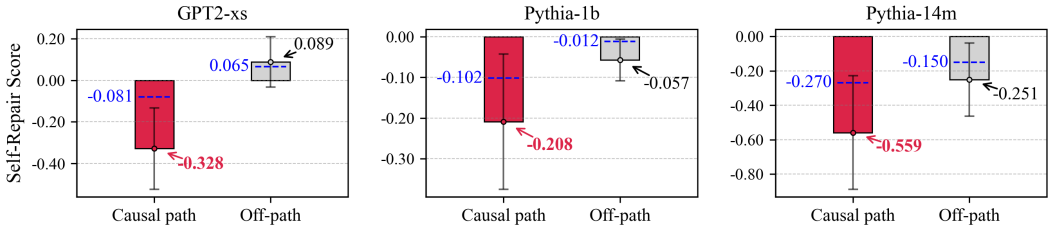
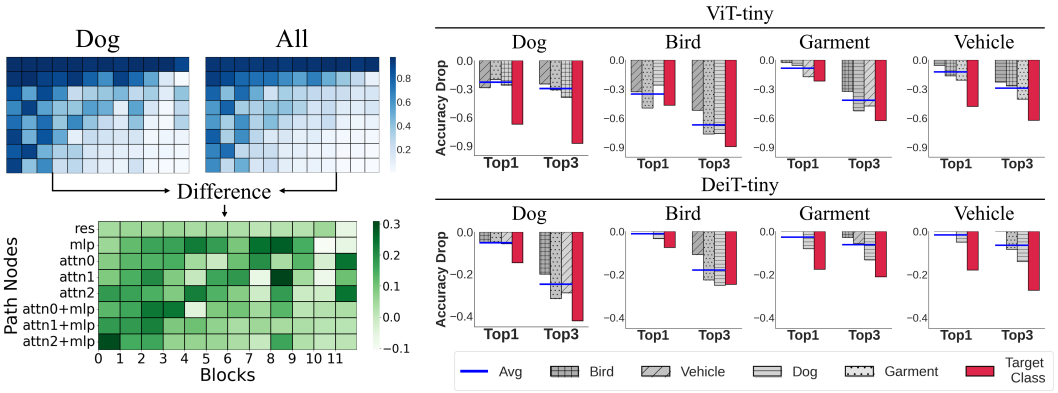


Figure 3: **Self-repair scores on causal path vs. off-path components.** Each bar shows the mean (dot with arrow) and standard deviation (error bar); medians are shown as blue dashed lines. Lower scores indicate less self-repair. Results are averaged over KNOWNS1000 and T-REX.



(a) (b)

Figure 4: **Causal paths uniquely activated for specific classes. (a)** Average causal path ratios for a target class (left), all classes (right), and their difference (bottom), highlighting class-specific paths. Here, res, mlp, and attn# indicate residual, MLP, and attention paths from head #, respectively. **(b)** Accuracy drop when ablating the most class-specific path, showing selective reliance by each class.

We find that self-repair occurs less frequently on the causal path. While self-repair is known to be highly noisy, as noted in [16], the results still show a clear difference: both the mean and median scores are consistently lower on the causal path than off it. This suggests that the causal path captures components essential to the decision and less reliant on backup mechanisms. In other words, the selected paths carry information not easily replaceable, underscoring their critical role for decision.

Class-specific causal subpaths play a functional role in predicting their respective classes Here, we aim to investigate whether the discovered causal paths contain class-wise causal nodes—nodes that are consistently utilized across samples within the same class group—and whether these nodes play a significant role in the model's classification decisions. To improve clarity, we first select four super-classes—dog, bird, garment, and vehicle—among the 1,000 ImageNet classes based on semantic similarity derived from WordNet. We then aggregated the causal paths extracted from individual samples and compiled statistics on the frequency of each subpaths' occurrence. By comparing these frequencies to the overall average across all samples, we identified causal subpaths that were significantly more active within specific super-classes (as shown in Figure 4-(a)). We refer to these as class-wise causal subpaths, hypothesizing that *they store key discriminative information relevant to their respective super-classes* due to their unusually high activation rates.

To validate this hypothesis, we intervene in the class-wise causal subpaths and measure the performance drop. If these nodes indeed encode class-specific information, their removal should lead to a greater accuracy drop within the corresponding super-class than in others. Figure 4-(b) clearly demonstrates this pattern. For instance, when the class-wise causal subpaths for the dog super-class were deactivated in a ViT-tiny model, the top-1 accuracy for dog samples decreased by approximately 44.7% more than that for other super-classes. Similar trends were observed across bird, garment, and vehicle classes, indicating that the proposed metric functions consistently across the model.

It is important to note that due to inherent semantic overlap among ImageNet classes, interventions on class-wise causal subpaths may still affect the logits of unrelated classes. Additionally, due to

	Hit. (↑)	Faith. (↑)	Spars. (↓)
NT_1	0.0000	0.0005	0.6571
$NT_{10\%}$	0.0000	0.0006	<u>0.5648</u>
ET_{all}	0.2079	0.2354	0.9806
ET_{cls}	<u>0.4808</u>	<u>0.4734</u>	0.9909
CPT	**0.9826**	**0.5466**	**0.8641**

Table 2: **Quantitative results (language).** Averaged over three models on two datasets; full results in Appendix.

	Hit. (↑)	Faith. (↑)	Spars. (↓)
NT_1	0.0105	0.0136	0.7276
$NT_{10\%}$	0.0078	0.0133	<u>0.0799</u>
ET_{all}	<u>0.4454</u>	<u>0.3166</u>	0.9999
ET_{cls}	0.2627	0.1832	0.9650
CPT	**0.9638**	**0.2991**	**0.7280**

Table 3: **Quantitative results (vision).** Averaged over two models on two datasets; see Appendix for details.

visual diversity within each super-class, turning off only a small number of subpaths may not entirely collapse performance. Nevertheless, the consistent and pronounced patterns observed across all super-classes suggest that our method effectively identifies causal subpaths that play a meaningful role in class-specific inference.

Quantitative results show our method yields reliable and faithful explanations Each value in Tables 2 and 3 represents the average score across models on two datasets. All methods are evaluated by pruning the model to retain only the paths identified by each method. We report three metrics: **Hit.** (hit rate) measures the proportion of cases in which the pruned model produces the same decision as the original; **Faith.** (faithfulness) quantifies the ratio of the original logit preserved after pruning; and **Spars.** (sparsity) denotes the proportion of model parameters retained by the identified path.

Our method (CPT) achieves a near-perfect hit rate, consistent with the theoretical guarantee in Theorem 2 that the identified paths are reliably causal. In contrast, existing methods show substantially lower hit rates, supporting our claim in Table 1 that while tracing is feasible with backward chaining, it is generally not reliable for identifying true decision paths. CPT also achieves the highest faithfulness, indicating that it preserves the model's original decision behavior more accurately than alternative methods. Notably, it does so while retaining significantly fewer parameters: whereas edge-level methods such as ET_{all} and ET_{cls} rely on nearly the entire model, CPT produces more faithful and compact explanations through substantially more efficient path selection.

4 Conclusion

In this paper, we presented an automated framework for tracing causal paths given a decision. We provide both theoretical analysis and empirical evidence showing that our method efficiently uncovers all causal paths responsible for a decision, with average-case polynomial-time complexity. Furthermore, we demonstrated that the identified causal paths (1) are less susceptible to self-repair effects, (2) reveal the structural grounds for subpaths uniquely activated for specific classes, and (3) yield more faithful and precise explanations than existing methods.

Limitations and Future Work. First, the identified causal paths are derived under the assumptions of our proposed framework and may not generalize under different assumptions. In particular, our unfolding procedure assumes uniform propagation of bias terms across all paths; however, accurately quantifying their individual contributions is non-trivial and remains an open direction for future work. Second, we acknowledge that our experiments were conducted on smaller models compared to state-of-the-art architectures. Although our method achieves polynomial-time complexity on average, large models may still incur prohibitive runtime in worst-case scenarios, and the reduced search space can remain sizable. Extending our minimality-based subset search to also prune supersets of non-causal subsets could mitigate this issue. Lastly, while our analysis focuses on structural mechanisms within the model, it opens avenues for future integration with feature attribution methods, potentially bridging structural and feature-level interpretability.

Despite these limitations, our work is the first to propose an efficient and reliable framework for tracing causal paths within transformer models for a given decision. We believe this represents an important step toward making transformers more transparent and robust in safety-critical domains, helping to prevent misuse and improve trust in deployment.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[2] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

[3] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

[4] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401, 2020.

[5] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

[6] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

[7] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022.

[8] Stefan Heimersheim and Jett Janiak. A circuit for python docstrings in a 4-layer attention-only transformer. In *Alignment Forum*, 2023.

[9] Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. *arXiv preprint arXiv:2309.16042*, 2023.

[10] Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

[11] Adithya Bhaskar, Alexander Wettig, Dan Friedman, and Danqi Chen. Finding transformer circuits with edge pruning. *Advances in Neural Information Processing Systems*, 37:18506–18534, 2024.

[12] Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 407–416, 2024.

[13] Lawrence Chan, Adria Garriga-Alonso, Nicholas Goldowsky-Dill, Ryan Greenblatt, Jenny Nitishinskaya, Ansh Radhakrishnan, Buck Shlegeris, and Nate Thomas. Causal scrubbing: A method for rigorously testing interpretability hypotheses. In *AI Alignment Forum*, volume 2, 2022.

[14] Michael Hanna, Ollie Liu, and Alexandre Variengien. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36:76033–76060, 2023.

[15] Neel Nanda, Senthooran Rajamanoharan, Janos Kramar, and Rohin Shah. Fact finding: Attempting to reverse-engineer factual recall on the neuron level. In *Alignment Forum*, page 6, 2023.

[16] Cody Rushing and Neel Nanda. Explorations of self-repair in language models. *arXiv preprint arXiv:2402.15390*, 2024.

[17] Joseph Y Halpern and Christopher Hitchcock. Actual causation and the art of modeling. *arXiv preprint arXiv:1106.2652*, 2011.

[18] Joseph Y Halpern. A modification of the halpern-pearl definition of causality. *arXiv preprint arXiv:1505.00162*, 2015.

[19] Algorithmic Research Group. gpt2-xs. `https://huggingface.co/AlgorithmicResearchGroup/gpt2-xs`, 2025. Accessed: 2025-05-16.

[20] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[22] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[23] Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[24] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, 2019.

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[26] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, stated in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, stated in the conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Yes, stated in the methodology and Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in Appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, stated in the experiments and the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Yes, stated in Appendix along with an anonymized code link.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Yes, stated in the experiments and the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in Appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Yes, stated in the experiments and the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, stated in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, we have identified no concerns.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, stated in the introduction and the conclusion.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work does not introduce or release any new models or datasets. Rather than posing risks of misuse, it contributes to preventing them by improving model transparency and interpretability.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use publicly available pretrained models (e.g., GPT2, Pythia) under their respective licenses. All models and tools used are properly credited and cited in the main paper. No modifications were made to their original distributions, and we adhered to their terms of use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release anonymized code in the Appendix. The code includes instructions for reproducing the main experiments.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This work does not involve crowdsourcing or any research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work does not involve research with human subjects and does not require IRB or equivalent approval.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We used an LLM solely for grammar correction.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.