FROM EMBEDDINGS TO ATTENTION HEADS: INPUT-AGNOSTIC TRACING OF EMBEDDING-MLP-ATTENTION CIRCUITS IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Input-agnostic mechanistic interpretability explains neural computation from model parameters without repeated forward passes. Existing input-agnostic methods still face a tradeoff: analyses of interactions between modules are mostly limited to toy transformers, while studies of standard LLMs usually focus on isolated modules, leaving key residual-stream interactions underexplored. We address this gap by introducing a task-conditioned zero-pass framework for tracing embedding→native gated-MLP-neuron→attention-head interactions in standard LLMs. Our method requires no task-example forward passes for activation/gradient collection. To our knowledge, prior input-agnostic work has not traced these interactions at this scale in a standard pretrained LLM. Experiments against input-dependent methods, including state-of-the-art causal path patching methods, show that our method achieves a highly favorable balance between causal effectiveness and computational efficiency.

1 INTRODUCTION

Mechanistic interpretability asks how a large language model (LLM) internally computes a prediction or behavior, rather than only describing what inputs and outputs are correlated with it. Existing mechanistic interpretability methods can be roughly divided into *input-dependent* and *input-agnostic* ones. Input-dependent approach studies a model on concrete inputs, typically by examining how internal activations change with the input or by intervening on those activations to test whether they are causally responsible for an output or intermediate behavior (Chen et al., 2025; Wang et al., 2023; Chen et al., 2026b; Meng et al., 2022; Chen et al., 2026a). Earlier approaches such as statistics (Antverg & Belinkov, 2022; Htut et al., 2019), visualization (Vig, 2019), and probing (Müller-Eberstein et al., 2022; Liu et al., 2026) can also be viewed as following this input-dependent scheme.

A primary limitation of input-dependent methods is computational cost: they require running the model on many input samples. Furthermore, patching-based methods are especially expensive because they often require multiple clean, corrupted, and patched forward passes (Wang et al., 2023), and sometimes backward passes (Nanda, 2023), even for a single example. In addition, these methods often rely on specialized input designs, such as corrupted or counterfactual text, and can be sensitive to input format, corruption scheme, and other evaluation hyperparameters. Recent work has shown that activation-patching conclusions can vary substantially with methodological choices such as the corruption method and evaluation metric (Zhang & Nanda, 2024), and that circuit-faithfulness conclusions can shift under seemingly small changes of ablation settings (Miller et al., 2024).

This motivates **input-agnostic** (also known as weight-only / zero-pass) interpretability. It seeks to uncover universal functional mechanisms within the model by analyzing parameter distributions and matrix properties, thereby eliminating the dependency on inputs and intermediate activations. Starting from the Transformer Circuits framework of Elhage et al. (2021), there has been an active line of research on analyzing model computation directly from learned weights: for example, Dar et al. (2023) projected transformer parameters into embedding space to interpret what tokens they operate on, Geva et al. (2022) viewed FFN layers as promoting concepts in vocabulary space, Elhelo & Geva (2025) inferred attention-head operations directly from their parameters, and Kamoda et al.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

(2025) analytically explained specific computations such as first-layer detokenization from weights alone.

However, existing input-agnostic work faces two complementary limitations. Recent work on standard language models has made analysis tractable by focusing on isolated modules, such as individual attention heads (Elhelo & Geva, 2025; Kamoda et al., 2025) or MLP layers (Geva et al., 2022). While useful for assigning functions to specific components, this module-centric perspective largely abstracts away the cross-module interactions through which many model behaviors may emerge. Another line of work explicitly study interactions: Transformer Circuits analyses introduced Q/K/V composition to characterize how attention heads compose with one another (Elhage et al., 2021). However, these analyses were developed in overly simplified settings, such as two-layer attention-only transformers. As a result, they provide valuable conceptual tools but do not readily scale to modern transformers with many layers, attention heads, and MLP blocks. This omission matters because preceding MLPs can be the strongest causal influencers of later attention heads, as we later verify.

We believe this bitter tradeoff is driven by two main difficulties. First, **MLPs introduce non-linear transformations** that are much harder to analyze statically than attention-head compositions. Early work such as Transformer Circuits (Elhage et al., 2021) largely sidestepped this difficulty by simplifying transformers into attention-only models; later input-agnostic work on standard language models (Elhelo & Geva, 2025; Geva et al., 2022), rather than fully modeling this non-linearity, often shifted toward analyzing single modules in isolation. Second, **studying weights alone** without considering the residual-stream directions they actually act on is often insufficient: a connection may look strong in the weight matrices, yet contribute little on the relevant activations.

To address these limitations, we propose an input-agnostic mechanistic interpretability method for efficiently identifying cross-module interactions that easily scales to large model architectures. Our method is built on a decomposition of MLP outputs into neuron-level vectors weighted by activation strengths. This directly incorporates MLPs into the analysis and grounds the method in the residual-stream directions that later attention heads use. Under this formulation, our framework can identify which neurons strongly affect a later attention head and which neurons are likely to activate at specific inputs in an input-agnostic manner. In particular, it traces interactions from *input embeddings*, at a granularity as fine as individual neurons, all the way to *target attention heads*. In this way, it addresses the two difficulties above together: it makes MLPs part of the analyzable mechanism, and it ties weights analysis to the actual residual-stream signals that those weights write and later heads read. We compared our method against several baselines(Antverg & Belinkov, 2022; Wang et al., 2023), including statistical activation-based neuron selection, gradient-based MLP attribution, and a tailored adaptation of the state-of-the-art causal mediation analysis, path patching. To evaluate discovered neurons causally, We further designed a perturb-then-restore-based benchmark. Experimental results show that our method substantially outperforms statistical and gradient-based baselines, while remaining competitive with path patching even when the latter is given much larger computational and sample budgets.

Overall, our contributions are three-fold. First, it opens a new regime of input-agnostic mechanistic interpretability: from toy models to standard large language models, from isolated modules to cross-module interactions, and from purely static weight analysis to residual-stream-aware analysis. Second, to our knowledge, it is the first input-agnostic method to trace an interaction circuit from attention heads all the way down to input embeddings in standard pretrained LLMs. Third, it achieves an exceptionally favorable balance between causal effectiveness and computational efficiency, remaining competitive with strong input-dependent tracing methods while requiring only negligible computation by comparison.

Our code is released anonymously at https://anonymous.4open.science/r/InputAgnosticMLPTrace-0849/.

2 Related Work

Our work is closely related to interpretability methods, which can be broadly categorized into two main paradigms: *input-dependent* and *input-agnostic*.

On the input-dependent side, methods analyze how models respond to concrete data samples. Early approaches often relied on observational signals: **visualization** methods identify key components by

directly observing their activation magnitudes (Li et al., 2016), and similarly, **gradient-based** methods determine component importance through the gradients computed during a backward pass (Tang et al., 2024). Additionally, **probing** methods test whether a linguistic property can be recovered from an internal representation by training a supervised classifier (Müller-Eberstein et al., 2022; Liu et al., 2026). To go beyond correlation, **patching-based** interpretability studies what internal components causally do by perturbing or restoring selected activations and measuring the resulting changes in model behavior (Wang et al., 2023; Nanda, 2023). Building upon these causal interventions, **automated circuit discovery** methods have recently emerged to systematically search and trace these functional sub-networks at scale (Conmy et al., 2023; Syed et al., 2024).

Input-agnostic methods, on the other hand, study model computation through mathematical formulations and weight matrices rather than interventions over concrete inputs. Elhage et al. (2021) introduced the circuit perspective and showed how transformer computation can be analyzed through compositions between modules, especially Q/K/V compositions between attention heads, though mainly in toy or attention-only settings. Much of the later input-agnostic literature on standard language models instead focused on analyzing single modules in isolation, such as attention heads (Elhelo & Geva, 2025; Kamoda et al., 2025; Olsson et al., 2022) or feed-forward layers (Geva et al., 2022). Different from them, we focus on *cross-module* analysis: rather than interpreting one head or one MLP layer in isolation, we study end-to-end input embedding→MLP neuron→attention head interactions in standard transformer language models.

For MLP nonlinearity and neuron-level analysis, previous work has approached these problems by modifying MLP structures or learning replacement feature bases, such as Bilinear MLPs (Pearce et al., 2025), transcoders (Dunefsky et al., 2024), and Sparse Feature Circuits (Marks et al., 2025). They're also either input dependant or operating at single module level. Our setting traces native gated-MLP neurons and their interactions with later heads.

3 METHODOLOGY

In this section, we introduce our input-agnostic framework for tracing input embedding → MLP → target attention head interactions. By decomposing MLP outputs into neuron-level contributors, the method breaks the barrier posed by MLP non-linearity and also identifies concrete contributors written into the residual stream. While previous methods mainly analyze neuron weights in isolation, our formulation follows how these neuron-level contributors can be read by later attention heads. In this way, it shifts input-agnostic interpretability from toy-model compositions or isolated-module analysis toward cross-module analysis in standard LLMs.

We begin by explaining why purely weight-based analysis can be unreliable, and why we need a formulation that is closer to the model's real computation. We then break down MLP computation into neuron-level contributions and derive an embedding-facing signal for which inputs are likely to activate a given neuron. Finally, we derive a neuron-pair composition score for how MLP neurons can theoretically contribute to the pre-softmax score of a target attention head.

A common pitfall of input-agnostic analysis is to study the weights alone and treat large norms or large matrix products as if they implied large functional influence. However, this can be misleading. For example, the matrix $\begin{pmatrix} 1000 & 0 \\ 0 & 1 \end{pmatrix}$ has a very large norm, but when applied to a vector $(0,1)^\top$ (this conceptually corresponds to a multiplication operation of a layer's weight on the residual stream), it only produces the identical transformation $(0,1)^\top$. Likewise, in attention, larger query/key norms do not necessarily imply a larger score. A simple example is

$$Q_1 = (1,-1), \qquad K_1 = (-1,1), \qquad \|Q_1\| = \|K_1\| = \sqrt{2}, \qquad Q_1 \cdot K_1 = -2,$$

while

$$Q_2 = \left(\frac{1}{2}, \frac{1}{2}\right), \qquad K_2 = \left(\frac{1}{2}, \frac{1}{2}\right), \qquad \|Q_2\| = \|K_2\| = \frac{\sqrt{2}}{2}, \qquad Q_2 \cdot K_2 = \frac{1}{2}.$$

Here the first pair has much larger norms, yet gives a smaller attention score. These examples suggest that the desired formulation should correspond to a real computation process, rather than only to static magnitudes. These two examples illustrate why static norm-based analysis can be unreliable.

3.1 BREAKING DOWN MLP COMPUTATION

The challenge is that real transformer computation is carried by the residual stream, whose values depend on concrete inputs and are therefore noisy from a purely input-agnostic perspective. We find that MLP decomposition helps bridge this gap. For the gated MLP structure considered here, the output can be written in matrix form as

$$\mathrm{MLP}(x) = W_{\mathrm{down}}\Big((W_{\mathrm{up}}x) \odot \mathrm{act_fn}(W_{\mathrm{gate}}x)\Big), \tag{1}$$

where $\odot$ denotes element-wise multiplication. This yields the neuron-wise decomposition

$$\mathrm{MLP}(x) = \sum_{i=1}^{d_{\mathrm{MLP}}} m_i v_i, \tag{2}$$

where a "neuron" i refers to one dimension of the intermediate gated activation vector. This dimension is associated with an input-side activation strength and an output-side value vector. Here v_i is the value vector associated with the i-th activation dimension, i.e. the i-th row of W_{down}, and

$$m_i = (W_{\mathrm{up}}^{(i)}x) \cdot \mathrm{act_fn}(W_{\mathrm{gate}}^{(i)}x) \tag{3}$$

is the corresponding activation strength. In this form, MLP output is explicitly decomposed into neuron-level vectors written into the residual stream.

The same decomposition also clarifies the role of MLPs in the residual stream. For token position t, the residual-stream vector entering a target attention head at layer ℓ_h can be decomposed as

$$x_t^{(\ell_h)} = e_t + \sum_{\ell<\ell_h} h_{\mathrm{Attn},t}^{(\ell)} + \sum_{\ell<\ell_h}\sum_{r=1}^{d_{\mathrm{MLP}}} m_{t,r}^{(\ell)} v_r^{(\ell)}, \tag{4}$$

where e_t is the input embedding, $h_{\mathrm{Attn},t}^{(\ell)}$ is the attention output at position t from layer ℓ, and $m_{t,r}^{(\ell)}v_r^{(\ell)}$ is the contribution of neuron r from the layer-ℓ MLP. The equation identifies three major writers to a later head's query/key inputs: embeddings, preceding attention blocks, and preceding MLPs. Module-level path patching in Appendix B finds that preceding MLPs cause the largest changes in target heads, further showing that the (yet underexplored) MLP neurons can be the strongest causal influencers of later attention heads.

Since our goal is to analyze how inputs can eventually influence a target attention head through MLP neurons, we next ask which inputs are likely to activate the neuron coefficient m_i in Equation 2, but without access to concrete inputs. To do so, we take the coefficient computation in Equation 3 and replace x with the full embedding matrix E, so that the same computation is evaluated over the entire vocabulary at once. This gives

$$g_i = (W_{\mathrm{up}}^{(i)}E) \odot \mathrm{act_fn}(W_{\mathrm{gate}}^{(i)}E), \tag{5}$$

where $g_i \in \mathbb{R}^{n_{\mathrm{vocab}}}$ gives a vocabulary-level score for neuron i. Intuitively, g_i describes which embeddings are likely to activate neuron i through the residual connection. We call g_i the **embedding-facing score**, and refer to its top-ranked words as the neuron's *activation lexicon*. This is the same formulation as Dar et al. (2023), applying learned MLP parameters directly to the embedding matrix. Despite following this established paradigm, to further ensure that the approximation error is acceptable, we conduct embedding facing-neuron activation correlation, layer-wise neuron-distribution, and clamp-layer analyses in Appendix C. We also test a more radical activation-lexicon-free alternative, which yields informative neuron-pair discoveries while remaining computationally efficient; see Appendix E.

3.2 MLP–ATTENTION HEAD COMPOSITION

We now turn to the target attention head. Let h be a target attention head in layer ℓ_h. Its pre-softmax attention score from source position s to destination position t is $x_s^\top W_{QK}^{(h)} x_t$, where x_s and x_t are the residual-stream vectors entering head h at the two token positions (we remove the layer index to avoid overcomplication).

Substituting the decomposed residual stream on both sides gives

$$\Big(e_s + \sum_{\ell<\ell_h} h^{(\ell)}_{\mathrm{Attn},s} + \sum_{\ell<\ell_h}\sum_{r=1}^{d_{\mathrm{MLP}}} m^{(\ell)}_{s,r} v^{(\ell)}_r\Big)^\top W^{(h)}_{QK}\Big(e_t + \sum_{\ell<\ell_h} h^{(\ell)}_{\mathrm{Attn},t} + \sum_{\ell<\ell_h}\sum_{r'=1}^{d_{\mathrm{MLP}}} m^{(\ell')}_{t,r'} v^{(\ell')}_{r'}\Big). \quad (6)$$

This shows that the target head's pre-softmax score is a sum of many interaction terms, including embedding-embedding, attention-attention, attention-neuron, and neuron-neuron terms. Among them, the neuron-neuron pair term takes the formj

$$m^{(\ell)}_{s,r} m^{(\ell')}_{t,r'} \left(v^{(\ell)}_r\right)^\top W^{(h)}_{QK} v^{(\ell')}_{r'}. \quad (7)$$

Since the input-side coefficients $m^{(\ell)}_{s,r}/m^{(\ell')}_{t,r'}$ in Equation 7 can be analyzed separately through the process introduced in Equation 5, in this subsection we focus on the rest of the Equation 7, which leads to the following score

$$B^{(h)}_{(\ell,r),(\ell',r')} = \left(v^{(\ell)}_r\right)^\top W^{(h)}_{QK} v^{(\ell')}_{r'}. \quad (8)$$

We refer to $B^{(h)}$ as the **MLP–attention composition score**. It should be interpreted as the *theoretical contribution of a neuron pair to the target head's pre-softmax attention score when the two coefficients are set to 1*. In other words, it measures how strongly the two neuron value vectors can contribute to the target head once they are activated, regardless of the actual inputs. By stacking all earlier-layer neurons, we obtain a composition matrix $B^{(h)} \in \mathbb{R}^{N_{<h}\times N_{<h}}$, where $N_{<h}$ is the total number of MLP neurons before layer ℓ_h.

Taken together, the embedding-facing score given in Equation 5 tells us which neurons are likely to activate at particular input positions, while the MLP–attention composition score in Equation 8 tells us which neuron pairs can strongly influence a target head once activated. This forms the core of our input-agnostic formulation for tracing input embedding→MLP→target attention head interactions. The task-specific use of these quantities for neuron selection and evaluation is introduced later in the experimental section. During real model inference, rotary position embeddings and RMSNorm also affect attention computation. In Appendix C, we evaluate RoPE across relative positions and derive the RMSNorm-adjusted composition, finding high selection stability across RoPE positions and preserved within-pair ranking under RMSNorm.

4 Experiments

In this section, we design evaluation frameworks and conduct experiments to answer the following research questions: **RQ1**: Can our proposed input-agnostic framework uncover faithful and practically useful mechanisms in real-world LLMs on established interpretability benchmarks? **RQ2**: What advantages do the key design choices of our framework (the end-to-end tracing of embedding → neuron → attention circuits, and the scalar neuron-composition formulation) provide compared with earlier static / weight-only methods? **RQ3**: Can our method make input-agnostic tracing a practical alternative to expensive input-dependent interpretability methods, especially in terms of the quality-efficiency tradeoff?

4.1 Basic Settings

To adress **RQ1**, We evaluate the proposed method under two benchmark task settings in interpretability: labeled syntactic dependency probing and indirect object identification (IOI). Dependency syntax represents a sentence as a set of labeled <*head-dependent*> word-to-word relations such as (*det* dependency) for determiner-constrained noun phrases like *the book*, *prep* dependency for verb-perposition compositions like *look at*, etc.; It is a standard formalism for languages in a syntactical view having wide downstream applications including relation extracting, sentiment analysis, prompting, and controllable generation (Wang et al., 2025; Fan et al., 2025; Yin et al., 2024; Liu et al., 2024). IOI task (Wang et al., 2023) asks the model to resolve which of two previously mentioned names is the correct indirect object in prompts such as: "When Mary and John went to the store, John gave a drink to ___." This task is a vital mechanistic interpretability task since it is one of the most well-known "in-the-wild" circuit discoveries with a concrete head-level circuit decomposition.

For labeled syntactic probing, we follow the dependency probing setting of Liu et al. (2026). Their parameter-free information-theoretic method estimates the mutual information (MI) between an attention head's pattern and dependency labels, and we reuse the resulting high-MI dependency heads as target heads in our experiments. The method is applied on the Universal Dependencies (UD) dataset (de Marneffe et al., 2021). Among dependency relations, we mainly focus on those involving functional words, such as *det*, *prep*, *cc*, and *aux*, because one side of such relations is comparatively well formed lexically.

For IOI, we use the indirect object identification task introduced by Wang et al. (2023). In particular, their mechanistic work discovered *Name Mover Heads*, a series of heads that attend to the earlier indirect-object name and copy it toward the output through the residual stream. We use *Name Mover Heads* as target heads. To locate these heads, we calculate head-recall, i.e., for a head, calculate the proportion that it looks at (has largest attention weight) on previous IO occurrence at each end position. For the corresponding role vocabulary, we use a list of common English first names.

For models, we used `OpenLLaMA-7b`. OpenLLaMa (Geng & Liu, 2023) is a family of LLaMa-styled LLMs trained using the RedPajama dataset (Computer, 2023), with varying sizes. For other settings like data examples and details of discovering target attention heads, please refer to Appendix A.1.

For the actual experiments, we instantiate the proposed framework in a task-specific way. We calculate the overlap between each neuron's activation lexicon, i.e. the top-ranked words under its embedding-facing score in Equation 5, and a task-specific vocabulary that is easy to compute. This overlap score serves as the **embedding-facing gate**, filtering out neurons unlikely to activate at the relevant input positions. We then apply the **head-composition gate** by examining the highest-scoring neuron pairs under the MLP–attention composition matrix in Equation 8 and counting how often each candidate neuron appears among these top-scoring pairs; this count serves as the neuron's composition frequency. Finally, we rank candidate neurons by multiplying the scores from the two gates. In this way, the selected neurons are required to be both likely to activate at the relevant positions and structurally capable of influencing the target attention head. Full operational details are given in Appendix A.2.

4.2 Baselines

We compare our method with four baselines: random neuron selection, probeless neuron selection, gradient attribution, and a tailored adaptation of path-patching baseline.

Random neuron selection. As a minimal control, we randomly select a set of MLP neurons with the same size as the neuron set discovered by our method or by each non-random baseline. This baseline ablates out the possibility that patching any sufficiently large collection of neurons can produce non-trivial restoration effects. It also controls whether a specific baseline is significant, i.e., whether it obtains obvious gains compared with a size-matched random neuron set.

Probeless. The probeless baseline selects neurons according to activation statistics, without training an auxiliary probe (Antverg & Belinkov, 2022). For each MLP neuron (ℓ, r), we compare its average activation at task-relevant source positions with its average activation at target or comparison positions. Let $a_{\ell,r}(x_s)$ and $a_{\ell,r}(x_t)$ denote the activation of neuron r in layer ℓ at the source and target token positions, respectively. The probeless score is

$$s_{\text{probeless}}(\ell, r) = \left| \mathbb{E}_{x_s} \left[a_{\ell,r}(x_s) \right] - \mathbb{E}_{x_t} \left[a_{\ell,r}(x_t) \right] \right|. \tag{9}$$

We then select the neurons with the largest values of $s_{\text{probeless}}(\ell, r)$. Despite its simplicity, this method can be surprisingly competitive for neuron detection, as evaluated by (Fan et al., 2023).

Gradient attribution. We also compare against a gradient-based attribution baseline. For a target attention head h, we define $F_h(x; s, t)$ as the attention score assigned by head h from the task-relevant source token s to the target token t. For each MLP neuron activation $a_{\ell,r}(x)$, the gradient attribution score is

$$s_{\text{grad}}(\ell, r) = \mathbb{E}_x \left[\left| \frac{\partial F_h(x; s, t)}{\partial a_{\ell,r}(x)} \right| \right]. \tag{10}$$

We additionally tried multiplying the gradient by the activation value of the neuron itself. This corresponds to a first-order Taylor approximation of the causal effect of removing that neuron on the target attention head, and shares the same principle as attribution patching (Nanda, 2023) However, we found it inferior to the direct gradient score, so we used the latter.

Adapted path patching. Finally, we compare our method with state-of-the-art causal patching-based circuit discovery methods. Here the choice of baseline matters. Our method traces an end-to-end composition of an embedding-MLP circuit and an MLP-attention circuit, each of which characterizes direct influence through direct or residual connections. Ordinary activation patching is less suitable for this purpose, because patching one activation can affect later modules, whose changed outputs then further alter downstream activations. The mixed and entangled effects makes it difficult to isolate the specific circuits that we aim to trace. We found that Path patching (Wang et al., 2023) fits this purpose better since it is designed to trace part-to-part localized influences. However, directly applying it at our neuron granularity introduces a serious computational problem: Its standard implementation requires many patched forward passes to identify one head responsible for a later head or logit. This becomes even more expensive at neuron granularity since the number of possible embedding-neuron / neuron-head compositions is far larger. We therefore use a tailored adaptation of path patching that first traces embedding→neuron influence, being able to identify a larger candidate set in a single stage, and then tests neuron→head composition on the reduced search space. This preserves the localized tracing spirit of path patching while making neuron-level analysis computationally feasible; full details are given in Appendix F. Appendix C verifies its stability across different hyperparameter settings, including Monte Carlo sample counts and candidate-set budgets.

4.3 Evaluation Framework

To evaluate the neurons discovered by different methods, we use a **neuron restoration patching** framework. Given an input example x, we first run a clean forward pass and cache the MLP neuron activations. We then construct a corrupted input $\tilde{x}$ by perturbing the input at task-relevant token positions, and run the model again. Finally, we perform a third forward pass on the corrupted input while restoring the activations of a selected neuron set $\mathcal{N}$ at the same task-relevant positions to their clean values. We refer to these three runs as the *clean*, *corrupted*, and *patched* runs, respectively.

Formally, let $m_{t,r}^{(\ell)}(x)$ denote the activation coefficient of neuron r in layer ℓ at token position t under input x. For a selected neuron set $\mathcal{N}$ and a task-specific position set $\mathcal{T}(x)$, the patched run applies the intervention

$$\mathrm{do}\Big(m_{t,r}^{(\ell)}(\tilde{x}) \leftarrow m_{t,r}^{(\ell)}(x)\Big), \qquad (\ell, r) \in \mathcal{N},\ t \in \mathcal{T}(x), \tag{11}$$

while leaving all other activations unchanged from the corrupted run. Intuitively, this asks whether the neuron set discovered by a method is sufficient to recover the original behavior once the relevant input-side signal has been disrupted.

We report the task-specific performance of the clean, corrupted, and patched runs and compare how strongly different neuron sets restore the original behavior. For dependency probing, we evaluate the UUAS and ULAS of the dependency trees probed from attention scores under the clean, corrupted, and patched runs. For IOI detection, we evaluate the target-name logit difference and the KL divergence of the answer-name token across the three runs. For all settings, we collect results averaged over five runs with different random seeds. The 95% confidence intervals of the estimated means are narrow, and the intervals for most method pairs are fully or nearly disjoint. We therefore report the mean values as representative outcomes.

4.4 Ablation and Efficiency Analysis

To address **RQ2**, we analyze the main design choices in the proposed framework and compare their performance and computational cost. We consider five ablation settings. **Attention-composition only** keeps only the head-composition gate and removes the embedding-facing gate, so neurons are selected only according to their composition frequencies, as introduced in Equation 16. **Embedding-facing only** keeps only the embedding-facing gate and removes the head-composition gate, so neurons are selected only according to their role-vocabulary overlap under the embedding-facing score, as defined in Equation 13. **Downwards tracing** follows the earlier implementation of our

Table 1: Main results on dependency probing and IOI. Results are averaged over five runs with different seeds. For dependency probing, we report undirected unlabeled attachment score (UUAS) and undirected labeled attachment score (ULAS) of probed dependency tree after restoring the neuron set selected by each method. For IOI, we report the restored target-name logit difference w.r.t. the corrupted run, and KL divergence w.r.t. the clean run.

	Dependency Probing		IOI	
Method	UUAS ↑	ULAS ↑	Δ Logit vs. Corrupt. ↑	KL Div. vs. Clean. ↓
Clean run	45.21	28.52	7.0528	–
Corrupted run	39.83	14.16	–	3.2478
Random	39.81	14.16	0.0419	3.2206
Probeless	40.97	16.28	0.1547	3.1107
Gradient attribution	39.72	15.52	0.0563	3.2331
Path patching	44.73	23.76	1.0134	2.8957
Ours	44.02	22.19	1.1868	2.5766

Table 2: Ablation results on dependency probing. Results are also averaged over five runs with different seeds.

Method / Setting	UUAS ↑	ULAS ↑
Embedding-facing only	42.52	20.20
Attention-composition only	43.76	21.47
One-side attention	42.42	20.51
Downwards tracing	44.38	22.61
ℓ_2-normalized composition	44.10	22.30
Original Setting	44.02	22.19

method: it first computes the composition matrix $B^{(h)}$ over all neurons before the target head, filters neurons according to head-composition, and only then reranks them by role-vocabulary overlap. **One-sided attention composition** replaces the proposed two-sided scalar neuron-composition score and subsequent composition frequency with a simpler one-sided vector norm-based metric. Instead of measuring pairwise contributions through the bilinear form in Equation 6, we project each neuron's value vector to the query or key side of the target head (e.g. $W_Q v_r^{(\ell)}$ or $W_K v_r^{(\ell)}$) and rank neurons by the norm of the resulting projected vector. This can be viewed as another form of weight-composition baseline of the kind commonly used in earlier static analysis (Elhage et al., 2021). **ℓ_2-normalized composition** replaces every value vector with $v_i/\|v_i\|_2$; this structural alternative is invariant to changes in the scale of neuron vectors. To address **RQ3**, we designed the computational cost analysis by monitoring the **total TFLOPs** required by our method and each baseline. This lets us evaluate whether our method is achieving a practical tradeoff between restoration quality and computational budget.

4.5 Results and Analysis

Table 1 shows three clear trends. First, our method, path patching, and the probeless baseline all outperform the random baseline on both tasks, indicating that the recovered neuron sets are not accidental. Among them, our method and path patching achieve the strongest restoration, restoring dependency performance much closer to the clean run. while gradient attribution remains close to random, suggesting that local gradient information is not a sufficiently reliable neuron-selection criterion, and might suffer from problems of locality under this neuron-level setting. Second, both mechanistic methods show a clear gap over the probeless baseline, and our method restores dependency performance much closer to the clean run. This suggests that the proposed framework is recovering neurons with stronger causal relevance than those obtainable from activation statistics alone. Finally, compared with path patching, our method remains highly competitive and marginally outperforms it on IOI, while staying close on dependency probing. Overall, these results show that the proposed input-agnostic framework is substantially stronger than the probeless baseline and competitive with a much more expensive input-dependent causal tracing baseline. These results provide a positive answer to **RQ1**, showing that our framework can uncover faithful and practically useful mechanisms on established interpretability benchmarks in standard large language models.

Moreover, we evaluate **factual tasks** with noisier vocabularies and attention relations (Appendix D.3). We further conduct dependency tracing across different model families and scales (Appendix D.4). Furthermore, Appendix E uses output-vocabulary projections to interpret the discovered neurons' high-level semantics and functions.

The ablation results in Table 2 support two main conclusions. First, removing either gate hurts performance: both the embedding-facing-only and attention-composition-only variants are weaker than the full method, showing that end-to-end tracing is not only a methodological spotlight of our framework. It is also a real source of performance gain. Second, the one-sided attention variant has worse performance, with a even larger gap, supporting our claim that norm-based static composition is less reliable than the proposed scalar two-sided formulation. The normalized variant slightly improves the original setting, showing that the framework remains effective under a scale-invariant composition. Downwards tracing is slightly stronger on dependency probing, where target heads are comparatively clean, while dual gating remains our default for tasks with noisier or more superposed head functions. These ablations provide a positive answer to **RQ2**, showing that both end-to-end tracing and the scalar neuron-composition formulation contribute materially to the effectiveness of the framework. Appendix D further checks our method's robustness to composition-frequency and task-vocabulary settings and generalization across tasks and model scales.

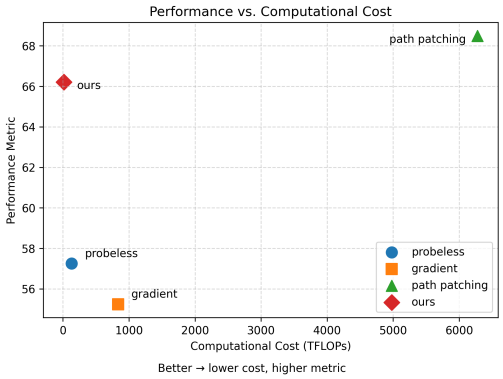

Figure 1: Figure of TFLOPS/metrics on dependency probing. x-axis stands for total TFLOPS required to run the method, while y-axis stands for the summation of UUAS and ULAS.

Figure 1 shows that our method achieves the best overall efficiency–performance balance. It remains competitive with path patching while requiring far less computation (14.53 TFLOPs for ours, compared with 130.57 for probeless, 834.25 for gradient attribution, and 6276.48 for path patching), showing that even simple input-dependent baselines are substantially more expensive than our method. This answers **RQ3** positively, showing that our method offers a strong quality–efficiency tradeoff relative to input-dependent baselines.

5 Conclusion

We presented an input-agnostic mechanistic interpretability framework for identifying input embedding→MLP→attention head interactions in standard transformer language models. By jointly modeling which neurons are likely to activate at relevant inputs and which neurons can strongly influence a target attention head, the framework provides an end-to-end trace from input embeddings, through MLP neurons, to later attention heads. Our experiments show that this formulation is not only conceptually meaningful but also practically effective: it substantially outperforms statistical and gradient-based baselines, while remaining competitive with a much more expensive causal tracing baseline based on path patching. The ablations further show that both gates are useful and that the proposed scalar two-sided formulation is stronger than simpler norm-based static alternatives, while the efficiency analysis shows that this better performance is achieved at even a far lower computational cost. More broadly, our work extends input-agnostic interpretability in three directions: from toy models to standard large language models, from isolated modules to cross-module interactions, and from purely static weight analysis to residual-stream-aware analysis. It shows that native gated MLPs can be incorporated into scalable input-agnostic cross-module tracing, allowing circuits to be traced from embeddings through MLP layers to later attention heads rather than being blocked at the MLP layers. We also verify the causal importance of MLP-mediated paths to attention heads, demonstrating the necessity and value of our method, and examine its approximation error and robustness across hyperparameter settings. We hope this provides a useful step toward scalable mechanistic analysis of realistic language models, and toward a better understanding of how higher-level behaviors are assembled across modules in modern LLMs.

Reproducibility Statement

Equations, selection procedures, evaluation protocols, computational costs, and implementation details are provided in the main paper and appendices. Anonymous code is available at the repository linked in the Introduction.

Ethics Statement

This work supports transparent analysis and diagnosis of language models. Like other interpretability tools, circuit-level analysis may also support capability-oriented model editing; we therefore emphasize scientific understanding and responsible use.

AI Use Statement

Large language models were used to aid and polish writing, assist with retrieval and discovery of related work, draft portions of the paper, provide feedback on experimental methodology, and assist in interpreting and presenting experimental results. The authors developed the research questions, proposed method, and experimental design, independently conducted the experiments, verified the cited literature and factual claims, reviewed and revised all AI-assisted text and analysis, and take responsibility for the manuscript.

References

Omer Antverg and Yonatan Belinkov. On the pitfalls of analyzing individual neurons in language models. In *International Conference on Learning Representations*, 2022.

Hang Chen, Jiaying Zhu, Xinyu Yang, and Wenya Wang. Rethinking circuit completeness in language models: And, or, and adder gates. In D. Belgrave, C. Zhang, H. Lin, L. Montoya, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 150511–150540. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/dd37fdb24a4e1cfa3ed5c247217a7394-Paper-Conference.pdf.

Hang Chen, Xinyu Yang, Jiaying Zhu, and Wenya Wang. Skill path: Unveiling language skills from circuit graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(36):30210–30217, Mar. 2026a. doi: 10.1609/aaai.v40i36.40271. URL https://ojs.aaai.org/index.php/AAAI/article/view/40271.

Hang Chen, Jiaying Zhu, Xinyu Yang, and Wenya Wang. CLUE: Conflict-guided localization for LLM unlearning framework. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=jtRYvazBWv.

Together Computer. Redpajama-data: An open source recipe to reproduce llama training dataset, 2023. URL https://github.com/togethercomputer/RedPajama-Data.

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.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. Analyzing transformers in embedding space. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16124–16170, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.893. URL https://aclanthology.org/2023.acl-long.893/.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. Universal Dependencies. *Computational Linguistics*, 47(2):255–308, June 2021. doi: 10.1162/coli_a_00402. URL https://aclanthology.org/2021.cl-2.11/.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable LLM feature circuits. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 24375–24410. Curran Associates, Inc., 2024. doi: 10.52202/079017-0768. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2b8f4db0464cc5b6e9d5e6bea4b9f308-Paper-Conference.pdf.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Amit Elhelo and Mor Geva. Inferring functionality of attention heads from their parameters. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17701–17733, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.866. URL https://aclanthology.org/2025.acl-long.866/.

Rui Fan, Shu Li, Tingting He, and Yu Liu. Aspect-based sentiment analysis with syntax-opinion-sentiment reasoning chain. In *Proceedings of the 31st international conference on computational linguistics*, pp. 3123–3137, 2025.

Yimin Fan, Fahim Dalvi, Nadir Durrani, and Hassan Sajjad. Evaluating neuron interpretation methods of nlp models. *Advances in Neural Information Processing Systems*, 36:75644–75668, 2023.

Xinyang Geng and Hao Liu. Openllama: An open reproduction of llama, May 2023. URL https://github.com/openlm-research/open_llama.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 30–45, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.3. URL https://aclanthology.org/2022.emnlp-main.3/.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R Bowman. Do attention heads in bert track syntactic dependencies? *arXiv preprint arXiv:1911.12246*, 2019.

Go Kamoda, Benjamin Heinzerling, Tatsuro Inaba, Keito Kudo, Keisuke Sakaguchi, and Kentaro Inui. Weight-based analysis of detokenization in language models: Understanding the first stage of inference without inference. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6339–6358, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.355. URL https://aclanthology.org/2025.findings-naacl.355/.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and understanding neural models in nlp. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 681–691, 2016.

Hongxu Liu, Xiaojie Wang, Jiashen Sun, Ke Zeng, and Wan Guanglu. Dual-stage multi-task syntax-oriented pre-training for syntactically controlled paraphrase generation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14215–14231, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.845. URL https://aclanthology.org/2024.findings-acl.845/.

Hongxu Liu, Jing Ma, Xiaojie Wang, Caixia Yuan, and Fangxiang Feng. An information-theoretic parameter-free bayesian framework for probing labeled dependency trees from attention score. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=q7raIuTQDK.

Samuel Marks, Can Rager, Eric Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 23888–23923, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/3ba4d47a83e498c2b1a0868cba20f6de-Paper-Conference.pdf.

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.

Joseph Miller, Bilal Chughtai, and William Saunders. Transformer circuit evaluation metrics are not robust. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=zSf8PJyQb2.

Max Müller-Eberstein, Rob van der Goot, and Barbara Plank. Probing for labeled dependency trees. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7711–7726, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.532. URL https://aclanthology.org/2022.acl-long.532/.

Neel Nanda. Attribution patching: Activation patching at industrial scale, 2023. URL https://www.neelnanda.io/mechanistic-interpretability/attribution-patching.

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.

Michael Pearce, Thomas Dooms, Alice Rigg, Jose Oramas, and Lee Sharkey. Bilinear MLPs enable weight-based mechanistic interpretability. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 47283–47310, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/7504142a20a3e1fe9dd7de42f475828c-Paper-Conference.pdf.

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*, pp. 407–416, 2024.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Wayne Xin Zhao, Furu Wei, and Ji-Rong Wen. Language-specific neurons: The key to multilingual capabilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5701–5715, 2024.

Jesse Vig. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th annual meeting of the association for computational linguistics: system demonstrations*, pp. 37–42, 2019.

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.

Lei Wang, Fei Wu, Xiaoqing Liu, Jilong Cao, Mingwei Ma, and Zhaoyang Qu. Relationship extraction between entities with long distance dependencies and noise based on semantic and syntactic features. *Scientific Reports*, 15(1):15750, 2025.

Wen Yin, Cencen Liu, Yi Xu, Ahmad Raza Wahla, Huang Yiting, and Dezhang Zheng. Synprompt: syntax-aware enhanced prompt engineering for aspect-based sentiment analysis. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (LREC-COLING 2024)*, pp. 15469–15479, 2024.

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.

A Additional Basic Settings

A.1 Additional Task and Data Details

For the dependency experiments, we use the same Universal Dependencies data split protocol as Liu et al. (2026). The input-dependent baselines are trained on 1,024 training samples and evaluated on a held-out validation split of approximately 1,700 samples. We mainly report dependencies involving functional words, including *det*, *prep*, *cc*, and *aux*, because these relations are both comparatively recoverable from attention heads and comparatively regular on one lexical side, which makes them suitable for our token-activation gate.

A concrete dependency example is the sentence

$$\text{“}\underline{The}_{\text{det}}\ \underline{book}_{\text{head}}\text{ was on the table.”}$$

Here the determiner The is the dependent token with label *det*, and book is its head noun. Another example is

$$\text{“She looked }\underline{at}_{\text{prep}}\ \underline{him}_{\text{head/context}}\text{ carefully.”}$$

where the preposition at occupies the functional-word side of the relation. These function-word positions are the side from which we construct role vocabularies for the token-activation gate.

For IOI, we use the synthetic template-based dataset of Wang et al. (2023). The prompts are instantiated from hand-written templates by filling named slots such as the indirect object, subject, place, and transferred object. In our experiments, the training and test splits are generated from different template pools to reduce template memorization; the training set contains 248 prompts and the test set contains 152 prompts. We use the same common-English-name pool as the role vocabulary for the name side of the circuit.

A concrete IOI example is:

$$\text{When }\underline{Mary}_{\text{IO}}\text{ and }\underline{John}_{\text{S}}\text{ went to the store, }\underline{John}_{\text{S2}}\text{ gave a drink to }\underline{\text{to}}_{\text{end}}$$

where $Mary$ is the indirect object (IO), $John$ is the first subject occurrence (S), $John$ is the second subject occurrence (S2), and to marks the final observed token position (end) from which the next-token prediction is made. The correct continuation is the earlier IO token, $Mary$, rather than the repeated subject name.

For dependency tracing, we follow the head-selection setting of Liu et al. (2026). Their parameter-free probing method estimates the mutual information between an attention-head variable $A^{(h)}$ and a dependency label variable Y,

$$I(A^{(h)};Y) = \int f(a,y)\log\frac{f(a,y)}{f(a)p(y)}\,da\,dy,$$

where the required distributions are estimated with kernel density estimators. Heads with high estimated mutual information are then selected as dependency-relevant heads, and the resulting posterior distributions are used to probe dependency trees from attention scores. We directly reuse the high-MI heads identified by this method as target heads in our experiments, rather than re-training a separate dependency probe.

For IOI target-head selection, we use the Name Mover Head criterion from Wang et al. (2023). Operationally, for each head we compute a role-pair recall score, e.g. for the canonical *end→IO* pattern:

$$\text{Recall}^{(h)}_{\text{end}\to\text{IO}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{1}\left[\arg\max_j a^{(h,n)}_{\text{end},j} = \text{IO}^{(n)}\right], \tag{12}$$

where $a^{(h,n)}_{\text{end},j}$ is the attention weight of head h from the end position to token j in example n. In the implementation, we also track related role pairs such as end $\to$ S2 and S2 $\to$ S, but the main IOI tracing experiments focus on the heads with high recall on the canonical Name Mover pattern.

The `open_llama_7b` experiments in this paper are run on standard causal language models of different scales. Unless otherwise stated, the same task definitions, target-head definitions, and role-vocabulary construction are kept fixed across model sizes, so that the comparison isolates the effect of the neuron-discovery method rather than changing the task itself. And for dependency probing, we adopted the same setting of (Liu et al., 2026): filter out attention heads by normalized MI, targeting 2000 heads for `open_llama_7b`.

We used publicly available code and model artifacts whose licenses permit research use. In particular, the IPBP codebase[1] is released under the MIT License, and the Easy-Transformer codebase[2] is also released under the MIT License. The OpenLLaMA-7B model[3] is released under the Apache 2.0 license. For syntactic experiments, we used the Universal Dependencies (UD) dataset. The portion we used is under Creative Commons licenses. We used these artifacts in accordance with their stated licenses.

A.2 Operational Details of Task-Specific Neuron Selection

The core methodology in Section 2 defines two complementary quantities: the embedding-facing score, which characterizes which input embeddings are likely to activate a neuron, and the MLP–attention composition score, which characterizes how strongly neuron value vectors can theoretically contribute to the pre-softmax score of a target attention head. In the actual experiments, we instantiate these two quantities through a two-gate selection procedure.

We begin with the **embedding-facing gate**. Many target attention patterns correspond to relations for which one side is lexically regular. We therefore construct a role vocabulary $\mathcal{V}_{\mathrm{role}}$ for that side of the relation. For dependency probing, this is the functional-word side of the dependency relation; for example, determiners for *det* and prepositions for *prep*. For IOI, it is the set of candidate person names on the indirect-object side. Let $\mathcal{L}_i$ denote the top-k_{lex} words in neuron i's activation lexicon, derived from its embedding-facing score g_i. We measure the role alignment of neuron i by

$$c_i = \frac{1}{|\mathcal{L}_i|} \sum_{w \in \mathcal{L}_i} \mathbf{1}[\, w \in \mathcal{V}_{\mathrm{role}} \,]. \tag{13}$$

This score measures the fraction of top-ranked activation words that belong to the role vocabulary. We then retain the top-k_{cand} neurons under c_i as a candidate set $\mathcal{C}$.

We next apply the **head-composition gate**. For a target attention head h, let $\mathcal{N}_{<h}$ be the set of MLP neurons from all layers before the target head. We restrict the composition matrix to the submatrix

$$B^{(h)}_{\mathcal{C}\cap\mathcal{N}_{<h},\,\mathcal{N}_{<h}}, \tag{14}$$

whose rows are indexed by candidate neurons that occur before the target head, and whose columns are indexed by all earlier-layer neurons. From this restricted matrix, we select the top-k_{pair} highest-scoring entries and denote their index set by

$$\mathcal{P}^{(h)}_{k_{\mathrm{pair}},\mathcal{C}} \subseteq (\mathcal{C} \cap \mathcal{N}_{<h}) \times \mathcal{N}_{<h}. \tag{15}$$

We then convert pairwise composition into a neuron-level score by counting how often each candidate neuron appears on the candidate side of these top-scoring pairs:

$$f_i^{(h)} = \sum_{(p,q)\in\mathcal{P}^{(h)}_{k_{\mathrm{pair}},\mathcal{C}}} \mathbf{1}[\, i = p \,], \qquad i \in \mathcal{C} \cap \mathcal{N}_{<h}. \tag{16}$$

Intuitively, neurons that appear more frequently in the top-scoring pairs are more likely to influence the target head once activated.

Finally, we combine the two gates by ranking candidate neurons with the multiplicative score

$$s_i^{(h)} = c_i \cdot f_i^{(h)}. \tag{17}$$

[1] `https://github.com/ChristLBUPT/IPBP`

[2] `https://github.com/redwoodresearch/Easy-Transformer`

[3] `https://huggingface.co/openlm-research/open_llama_7b`

This score enforces the conjunctive logic of our method: a neuron should count as a valid contributor only if it is both likely to activate at the relevant positions and structurally capable of influencing the target attention head. The first factor is provided by the embedding-facing gate, and the second by the head-composition gate.

In the implementation used in this paper, the procedure is ordered for efficiency. We first compute the embedding-facing score for all neurons and select a top-k_{cand} candidate pool. We then compute the head-composition frequencies only on this reduced set. This ordering is computationally advantageous because embedding-facing scores can be precomputed once for a fixed model and task, while the restricted composition search is much smaller than evaluating the full all-neuron pair space. We set $k_{\text{lex}} = 50$, $k_{\text{cand}} = 3200$, $k_{\text{pair}} = 50$, and finally select 200 neurons ranking by $s_i^{(h)}$.

We ran our experiments on a computational server using Nvidia A100-40G graphcs cards and AMD EPYC 7502 CPU, with 512 GB Memory and several SSD arrays. The computation time of our method is 7 minutes for pre-calculating embedding-facing score and < 3 minutes for a single label (like *det* in dependency and *IO* in IOI detection), while the baselines generally cost more time, ranging from 10 minutes to several hours.

Despite its effectiveness, we still think our current framework has the following limitations: It focuses on one specific cross-module interaction family, namely input embedding→MLP→attention head, and has not yet been extended to other interaction patterns such as attention→MLP, MLP→MLP, or longer multi-hop circuits. In addition, the current default configuration is designed to favor computational efficiency, and the ablation results suggest that stronger but more expensive variants may further improve performance, so the full efficiency-performance frontier remains to be explored. Finally, our evaluation mainly focuses on restoration-based causal testing, which is well suited for comparing neuron sets across methods, but does not exhaust all possible notions of mechanistic faithfulness.

B Causal Influence of Preceding Modules

To compare the three types of writers in the residual-stream decomposition, we conduct module-level path patching across the dependency relations. We separately patch the input embedding, preceding attention outputs, and preceding MLP outputs, then measure the resulting change in the target attention head.

Table 3: Causal effects of preceding module types on target attention heads, averaged across dependency relations.

Patched source	Mean absolute output change	Attention KL
Input embedding	0.0082	0.0568
Preceding attention	0.0351	0.0926
Preceding MLP	**0.0541**	**0.2648**

Preceding MLPs produce the largest change under both measures, exceeding the effects of preceding attention blocks and input embeddings. This verifies that MLPs can be the strongest causal influencers of the later attention heads studied here and strengthens the motivation for explicitly tracing MLP-mediated paths.

C Approximation Error and Stability Analysis

C.1 Embedding-Facing Score Validation

We directly measure the Pearson correlation between Equation 5 and neuron activations on natural occurrences of the corresponding tokens. We use text8 and WikiText-103, which are not targeted to the dependency phenomena studied in the main experiments. Text8 is not segmented into sentences, so we divide it into fixed-length sequences. This pretraining-like segmentation further breaks sentence-level syntactic regularity. Natural corpora also contain many tokens to which the task-facing neurons

selected by our method are only weakly responsive, so we expect substantially more conservative results under this noisy distribution.

Table 4: Pearson correlation between embedding-facing scores and measured neuron activations.

Dataset	All neurons	Dual gate	Embedding-facing only
text8	0.100	0.408	0.537
WikiText-103	0.094	0.389	0.498

Dual-gate neurons reach approximately four times the all-neuron correlation on both datasets, while embedding-facing-only neurons exceed five times the all-neuron average. The result is consistent across both corpora and selection variants. Considering the noisy vocabulary distribution and the disrupted syntactic structure in text8, the selected-neuron correlations lie comfortably above the conservative all-neuron reference level, supporting the reliability of the embedding-facing score.

C.2 Neuron Layer Distribution and Clamp-Layer Analysis

We first examine how the neurons selected by the complete method are distributed across layers. As Table 5 shows, lower layers contribute more selected neurons, with layers 0 and 1 providing the largest shares.

Table 5: Layer distribution of neurons selected by the complete method.

Layer	0	1	2	3	4	5	6	7	8	9
Selected	122	140	79	25	18	15	31	18	24	16

More interestingly, we find that higher-layer neurons compose more tightly with later heads. Averaging $B^{(h)}$ over all heads, dependency labels, and other-side neurons, the mean head-composition score increases from -8.6904 at layer 0 to 0.2759 at layer 5. This might indicate an internal repair mechanism: higher-layer neurons' stronger composition with later heads can compensate for their less direct alignment with input embeddings. The repair mechanism might also explain why the downwards-tracing and composition-only ablation variants remain effective. We therefore analyze the layer distributions of neurons found by these two variants. Across layers 0–9, downwards tracing selects 109, 98, 30, 25, 40, 35, 25, 20, 14, and 23 neurons, while composition-only selection gives 45, 65, 48, 47, 38, 39, 38, 45, 41, and 27. Compared with dual-gating, both distributions are less skewed toward the earliest layers and place more mass on higher layers, where neurons internally have stronger compositions with later heads.

Furthermore, in order to analyze the effect of artificially removing lower-layer neurons, we restrict complete discovery to progressively deeper search ranges. This gives 42.0/18.9 UUAS/ULAS for layers ≥ 2, 41.0/16.4 for layers ≥ 4, and 40.7/15.7 for layers ≥ 8. Many target heads are themselves shallow or intermediate, so these cutoffs simultaneously narrow the eligible neuron and target-head ranges. Complementarily, embedding-facing-only selection from individual layers remains stable: UUAS ranges from 38.9 to 39.9 and ULAS from 14.1 to 15.1 over layers 2–16.

C.3 Effect of Rotary Position Embeddings

For relative position $\Delta = n - m$, the RoPE-aware composition is

$$B_{ij}^{(h,\Delta)} = v_i^\top (W_Q^h)^\top R_\Delta W_K^h v_j. \tag{18}$$

The dependency and IOI tasks studied here place most attention relations over short- and medium-range spans. Therefore, we recompute selected neurons for $\Delta = 0, \ldots, 5$, which covers most attention relations used in these tasks. The pairwise IoU matrix is:

The mean IoU is 0.928 and every comparison exceeds 0.85, showing high stability across the evaluated relative-position range.

Table 6: IoU between neuron sets selected using different relative positions.

Δ	0	1	2	3	4	5
0	1.0000	0.9664	0.9219	0.8795	0.8685	0.8700
1	0.9664	1.0000	0.9432	0.8947	0.8785	0.8725
2	0.9219	0.9432	1.0000	0.9431	0.9207	0.9014
3	0.8795	0.8947	0.9431	1.0000	0.9650	0.9288
4	0.8685	0.8785	0.9207	0.9650	1.0000	0.9624
5	0.8700	0.8725	0.9014	0.9288	0.9624	1.0000

C.4 Effect of RMSNorm

For RMSNorm,

$$\mathrm{RMSNorm}(x) = D_\gamma x/\sigma(x). \tag{19}$$

In LLaMA's pre-norm architecture, residual components accumulate additively before normalization. Writing $x_t = \sum_r c_{t,r}$ and absorbing D_γ into the adjacent projections gives

$$\frac{1}{\sigma_s \sigma_t} \sum_{r,r'} c_{s,r}^\top \widetilde{W}_{QK}^{(h)} c_{t,r'}. \tag{20}$$

The remaining source- and destination-dependent factors are shared by all component pairs for the same token pair. This means that, after applying our decomposition, the within-pair ranking of component-pair contributions does not change.

C.5 Stability of the Adapted Path-Patching Baseline

We vary both the Monte Carlo sample count and the Stage-1 candidate-set size to check whether the adapted baseline is sensitive to either computational budget.

Table 7: Adapted path-patching stability across Monte Carlo budgets.

Monte Carlo samples	UUAS	ULAS
128	44.3	22.8
256	44.2	22.7
512	44.2	22.7

Table 8: Adapted path-patching stability across Stage-1 candidate-set budgets.

Candidate-set size	UUAS	ULAS
800	44.8	23.6
3,200	44.7	23.8
6,400	44.4	23.1

The candidate-size sweep uses fewer evaluation examples because the larger settings require substantially more computation, so its absolute values should be compared within that sweep. Both tables show that the adapted baseline operates in a stable hyperparameter region.

D Robustness and Generalization

D.1 Varying Selection Hyperparameters

The embedding-facing gate produces a candidate-neuron set $\mathcal{C}$, while k_{pair} determines how many of the highest-scoring composition entries are used to calculate each candidate neuron's composition frequency. Because $|\mathcal{C}|$ varies across tasks and target heads, we set $k_{\mathrm{pair}} = |\mathcal{C}| \times$ multiplier so that the composition-search budget grows proportionally with the candidate set. Dependency performance remains nearly unchanged across three orders of magnitude:

Table 9: Robustness to the composition-frequency multiplier.

Multiplier	UUAS	ULAS
1	43.8	21.9
50	44.0	22.2
100	44.0	22.2
1000	43.9	22.1

These results show that our method is robust to the composition-frequency multiplier on dependency tracing.

D.2 Corrupting Task Vocabulary

Furthermore, to analyze our method's vulnerability to incomplete or corrupted task vocabularies, we randomly remove 20% of the task vocabulary or add 20% irrelevant tokens.

Table 10: Robustness to task-vocabulary perturbation.

Vocabulary setting	UUAS	ULAS
Remove 20%	42.4	20.6
Add 20% irrelevant tokens	44.0	22.1

Performance remains effective under both perturbations and is essentially unchanged when irrelevant tokens are added.

D.3 Factual Recall

We conduct experiments on the factual-recall dataset used by Meng et al. (2022), specifically its `known_1000` set. We treat subject tokens as the embedding-facing side and trace heads attending from the final prediction position to the subject. This task has a broad subject vocabulary and noisier attention patterns that can mix subject-to-output and relation-to-output attention. We broadly categorize the dataset by subject type and use its predominant category, person names, as the task class. Even within this category, many neurons have non-zero overlap between their embedding-facing lexicons and the subject vocabulary, making the setting especially challenging for our vocabulary-relying method.

Table 11: Results on factual recall.

Method	Logit difference ↑	KL divergence ↓
Ours	0.4746	1.7043
Path patching	0.3048	1.7242

Our method remains stronger than path patching under both factual-recall metrics.

D.4 Dependency Tracing Across Model Families and Scales

Furthermore, to evaluate generalization across model scales, we repeat dependency tracing on OpenLLaMA-3B. Our method achieves 44.8/16.1 UUAS/ULAS, approximately similar to the 44.02/22.19 obtained in the original OpenLLaMA-7B setting in Table 1.

We also evaluate Qwen3-8B, where the clean run achieves 36.1/17.1 UUAS/ULAS and corrupt-then-restore using neurons discovered by our method achieves 37.5/16.1. More interestingly, while ULAS remains close to the clean score, UUAS improves by 1.4 points. This suggests that selected-neuron restoration can sometimes produce a refinement effect on unlabeled dependency structure when the clean probing quality is relatively weak.

E VOCABULARY-FREE DISCOVERY AND NEURON INTERPRETATION

E.1 VOCABULARY-FREE MECHANISM DISCOVERY

In order to further explore whether our method can produce informative neuron-pair findings when no additional prior task knowledge is available, we remove task-dependent head selection and vocabularies while retaining our original computational formulations. The process is as follows:

1. For any head h, we compute its composition scores with every eligible pair of preceding MLP neurons using the same formulation as Equation 8:
$$S_{r,r'}^{(h,\ell,\ell')} = \left(v_r^{(\ell)}\right)^\top W_{QK}^{(h)} v_{r'}^{(\ell')}, \qquad \ell, \ell' < \ell_h. \tag{21}$$
2. We then calculate a readability score for every neuron's embedding-facing lexicon. Let a_{it} be neuron i's positive embedding-facing score for token t, and T_i its top-k tokens. We measure top-k energy concentration as
$$C_i = \frac{\sum_{t\in T_i} a_{it}^2}{\sum_{t\in\mathcal{V}} a_{it}^2 + \epsilon}, \tag{22}$$
and embedding concentration around the weighted centroid as
$$S_i = \sum_{t\in T_i} w_{it} \cos(\hat{e}_t, \mu_i), \tag{23}$$
where w_{it} is the normalized embedding-facing score and μ_i is the weighted centroid of the normalized top-token embeddings. Their percentile ranks are combined into a readability score.
3. We then filter each head's highest-ranked composition pairs whose two neurons both belong to the highest-readability neuron set. Computing composition scores first and filtering readable pairs afterward is unbiased and effectively avoids cherry-picking or favoring readable neurons during the composition search.
4. We inspect the resulting shortlist. Some example mechanisms are shown in Table 12.

Our method's efficiency advantage also remains when task-conditioned vocabulary restrictions are removed: computing the composition and readability scores for the first 16 layers requires approximately 0.5 A100 GPU-hours.

Table 12: Examples found without task-specific heads or role vocabularies.

Key-side lexicon	Query-side lexicon	Representative construction
take/taking/took	advantage/benefit/edge	take advantage
not	anything/anyone/anywhere	not see anything
might/may/would/could	anything/something/anywhere	could be anywhere
why/what	wouldn't/couldn't/didn't	Why wouldn't...?
didn't/couldn't/wasn't	know/understand/known	didn't know
November/September/June	five/seven/six/sixth	November 5

Taken together, these automatically discovered examples span logical, epistemic, temporal, and other relations, illustrating that the vocabulary-free procedure can uncover a broad and heterogeneous range of neuron-pair mechanisms.

E.2 OUTPUT-SIDE INTERPRETATION

To obtain a high-level interpretation of the semantics and functionality of the discovered neurons, we project each selected neuron's W_{down} row into the output vocabulary and inspect the highest-ranked tokens.

The output vocabularies consistently connect dependency-selected neurons to relation-specific continuations: determiner neurons promote nominal or quantified heads, preposition and coordination

Table 13: Representative output projections of selected neurons.

Relation	Promoted tokens	Interpretation
det	member, spokeswoman, colleague	nominal heads
det	seconds, weeks, months, years	quantified nominal heads
prep	accordance, relation, terms	relational heads
cc	consequently, thereby, therefore	discourse-linked continuations
aux	be, make, have, do, provide	verbal heads

neurons promote relational or discourse-linked continuations, and auxiliary neurons promote verbal heads. These projections provide a high-level semantic interpretation of the residual-stream directions written by the discovered neurons.

F Details of the Adapted Path-Patching Baseline

Path patching replaces part of a model's forward pass with activations from another input in order to measure the causal effect of a specific computational path. Given a clean input x and a corrupted input $\tilde{x}$, standard activation patching restores or corrupts activations at selected locations and observes the resulting change in a target behavior. However, ordinary activation patching does not distinguish direct from indirect effects: after a component is patched, the intervention may propagate through many subsequent modules. Path patching addresses this issue by restricting the intervention to specified paths, thereby estimating more direct causal influence.

The overall logic of our path patching method is as follows: Given a clean input x and a corrupted input $\tilde{x}$, we first cache clean activations from x, then run $\tilde{x}$ while patching module outputs back to their clean values and recording the resulting changes in MLP-neuron activations. These activation differences provide a more localized signal of how embedding corruption perturbs later neurons, before the perturbation is further mixed by other modules. We use this signal to rank a larger candidate set, then apply Monte Carlo subset sampling and path patching on the selected neurons to identify subsets that compose strongly with the target head. In this way, the procedure screens embedding→neuron influence efficiently and then tests neuron→head composition on a much smaller search space

Our setting differs from the original path-patching setup in two ways. First, our method aims to trace a second-order circuit consisting of two composed subcircuits:

$$\text{Input Embedding} \rightarrow \text{MLP Neuron} \rightarrow \text{Target Attention Head}. \tag{24}$$

The first subcircuit captures the direct effect of perturbing the input embeddings on MLP neurons, while the second captures the direct effect of these MLP neurons on the target attention head through direct or residual connections. Second, our analysis is performed at MLP-neuron granularity. Since a transformer contains far more MLP neurons than attention heads, exhaustively applying the original path-patching paradigm to every neuron-level path would require an impractical number of forward passes.

To make path patching feasible in this heterogeneous and neuron-level setting, we adapt the procedure as follows. Let x be the clean input and $\tilde{x}$ be the corrupted input. We first run a clean forward pass and cache the relevant module activations:

$$\mathcal{C} = \{z_m(x) : m \in \mathcal{M}\}, \tag{25}$$

where $z_m(x)$ denotes the output of module m, and $\mathcal{M}$ includes the attention and MLP modules involved in the intervention.

Next, we run the model on the corrupted input $\tilde{x}$. During this corrupted forward pass, we patch the output of each attention or MLP module back to its clean value from $\mathcal{C}$, while recording the distorted activations of individual MLP neurons. This isolates the direct effect of the corrupted input embeddings on MLP neurons, because other module outputs are prevented from freely propagating the corruption. For neuron (ℓ, r), we denote the recorded directly corrupted activation as $a^{\text{emb}}_{\ell,r}(\tilde{x})$, and define its embedding-induced distortion by

$$\Delta a^{\text{emb}}_{\ell,r} = a^{\text{emb}}_{\ell,r}(\tilde{x}) - a_{\ell,r}(x). \tag{26}$$

This quantity measures how strongly neuron (ℓ, r) is directly affected by the embedding perturbation.

We then form a larger candidate pool by selecting the top-k neurons with the largest embedding-induced distortions:

$$\mathcal{N}_{\text{emb}}^{(k)} = \text{TopK}_{(\ell,r)}\left(\left|\Delta a_{\ell,r}^{\text{emb}}\right|\right). \tag{27}$$

This step ranks candidate neurons in a single forward pass, avoiding the need to test all neurons individually with separate interventions.

If the candidate pool is still larger than the desired neuron set size, we perform Monte Carlo subset sampling. At sampling step j, we randomly select a subset

$$\mathcal{S}^{(j)} \subset \mathcal{N}_{\text{emb}}^{(k)}, \qquad |\mathcal{S}^{(j)}| = k_{\text{sub}}. \tag{28}$$

We then run a patched forward pass where neurons in $\mathcal{S}^{(j)}$ are patched to their directly corrupted values $a_{\ell,r}^{\text{emb}}(\tilde{x})$, while MLP modules that do not contain these neurons and unrelated attention head outputs are kept at their clean values. In this way, the sampled neurons transmit the embedding perturbation toward the target head, but unrelated indirect pathways remain suppressed.

For the target attention head h, we additionally separate the query and key input sides. When evaluating the effect through the query side, we allow the query input of h to receive the corrupted output induced by $\mathcal{S}^{(j)}$, while patching the key input to its clean value. Conversely, when evaluating the key side, we allow the key input to receive the corrupted output while keeping the query input clean. Denoting the two receiver sides by $r \in \{Q, K\}$ and the opposite side by $\bar{r}$, the intervention is abstractly written as

$$z_{h,r} \leftarrow z_{h,r}^{\text{corr}(\mathcal{S}^{(j)})}, \qquad z_{h,\bar{r}} \leftarrow z_{h,\bar{r}}^{\text{clean}}. \tag{29}$$

This isolates the extent to which the selected MLP neurons mediate the corrupted embedding information into one side of the target attention computation.

Each sampled subset is scored by its effect on the target attention behavior. Let $F_h(x; s, t)$ denote the attention score assigned by target head h from source token s to target token t. The effect of subset $\mathcal{S}^{(j)}$ can be measured as

$$J(\mathcal{S}^{(j)}) = \left| F_h^{\text{patched}(\mathcal{S}^{(j)})}(x; s, t) - F_h(x; s, t) \right|, \tag{30}$$

or by the corresponding restoration or corruption metric used in the main experiment. The final adapted path-patching baseline selects the sampled subset with the largest estimated effect:

$$\mathcal{S}^* = \arg\max_{\mathcal{S}^{(j)}} J(\mathcal{S}^{(j)}). \tag{31}$$

This adaptation preserves the causal-mediation motivation of path patching while making it applicable to our setting. It handles heterogeneous modules by explicitly decomposing the circuit into input-embedding-to-MLP-neuron and MLP-neuron-to-attention-head parts. It also avoids the intractability of exhaustive neuron-level path patching by tracing direct influence upward from corrupted embeddings: neurons can first be ranked in one forward pass, and the remaining combinatorial search is approximated through Monte Carlo subset sampling.