Navigating by Old Maps: The Pitfalls of Static Mechanistic Localization in LLM Post-Training

Hang Chen
College of Computing and Data Science
Nanyang Technological University
chen.hang@ntu.edu.sg

Jiaying Zhu
School of Computer Science and Engineering
The Chinese University of Hong Kong
jyzhu24@cse.cuhk.edu.hk

Hongyang Chen
Shaanxi Co., Ltd(Xi'an 710077, China)
China Mobile Group
chenhongyang@sn.chinamobile.com

Hongxu Liu
College of Computing and Data Science
Nanyang Technological University
hongxu001@e.ntu.edu.sg

Xinyu Yang*
School of Computer Science and Technology
Xi'an Jiaotong University
yxyphd@mail.xjtu.edu.cn

Wenya Wang†
College of Computing and Data Science
Nanyang Technological University
wangwy@ntu.edu.sg

Abstract

The "Locate-then-Finetune" paradigm is a predominant LLM post-training strategy that localizes critical paprameters via mechanistic interpretability for targeted fine-tuning while freezing remaining parameters. However, whether localized parameters faithfully represent target mechanisms, whether they evolve during fine-tuning, and whether static snapshots can genuinely guide dynamic optimization remain unexplored—a critical gap stemming from incomplete mechanistic interpretability. To investigate this, we systematically track the structural evolution of Transformer circuits throughout the supervised fine-tuning (SFT) process, revealing the underlying dynamics of task mechanisms. To get the complete observation, we introduce three novel metrics—Circuit Distance, Circuit Stability, and Circuit Conflict—to analyze circuit evolution across three dimensions: neural migration, semantic stability, and cross-task interference. Our empirical results reveal that circuits inherently exhibit "Free Evolution" during parameter updates. Consequently, static mechanisms extracted from current states inevitably suffer from temporal latency, making them fundamentally inadequate for guiding future states. Moreover, by deconstructing the "illusion of effectiveness" in existing methods, this work underscores the necessity of "foresight" in mechanistic localization and proposes a predictive framework for future research. Our code is available at https://github.com/Zodiark-ch/MechLocalization.

1 Introduction

The "**locate-then-finetune**" paradigm has recently emerged as a prominent post-training strategy for LLMs. By confining parameter updates to localized critical parameters, this paradigm is widely adopted across diverse downstream applications, including model unlearning [1, 2], knowledge editing [3, 4], and safety alignment [5].

*Corresponding author
†Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Specifically, this paradigm operates in two sequential steps. First (**Locating**), before fine-tuning, it leverages mechanistic interpretability to identify the minimal parameter space responsible for the target task within the pre-trained model—a process known as **Mechanistic Localization**. Second (**Finetuning**), all non-critical parameters are frozen, and target-specific fine-tuning is exclusively conducted on the localized parameters. For example, when aligning a model to a target task A, the system first localizes the specific components governing task A in the initial parameter state, and subsequently updates only these components. Substantial empirical evidence demonstrates that the locate-then-finetune paradigm not only enhances fine-tuning efficiency and target performance, but also prevents catastrophic forgetting on pre-existing capabilities (hereafter referred to as **pervasiveness tasks**) [6].

However, recent studies show that Mechanistic Localization often lacks completeness [7] and exclusiveness [8], raising a critical question: acts it merely as a "placebo"? Specifically, **can a static snapshot of Mechanistic Localization from pre-tuning model genuinely guide the dynamic process of future parameter updates?** As illustrated in Figure 1, if full-parameter SFT causes the target task's critical components to shift from B_1, B_2 to A_1, A_2, localization based solely on pre-tuning parameters will prematurely freeze A_1 and A_2. Therefore, it remains unclear whether Mechanistic Localization genuinely prevents conflicts or improperly constrains the target mechanism's natural evolution. To address this, we break down this problem into two specific research questions:

- **RQ1**: Without localization, do the critical components of the target skill change during SFT? If so, how do they evolve?
- **RQ2**: Does the Locate-then-Finetune paradigm truly improve performance on the target task and mitigate conflicts with pervasiveness tasks?

In this paper, we employ **SFT** as the representative of post-training process to intuitively observe the evolutionary process, defining the **component**[3] as the minimal update unit. For Mechanistic Localization, we adopt **circuit discovery** [9, 10], which comprehensively captures overarching mechanisms and inter-component connections. To address **RQ1**, we introduce **circuit distance** to quantify critical component migration, and **circuit stability** to evaluate the model's mastery of the task mechanism. For **RQ2**, alongside direct performance observation, we propose **circuit conflict** to measure how effectively localization prevents mechanistic clashes between the target and pervasiveness tasks. Through extensive experiments, we draw the following conclusions:

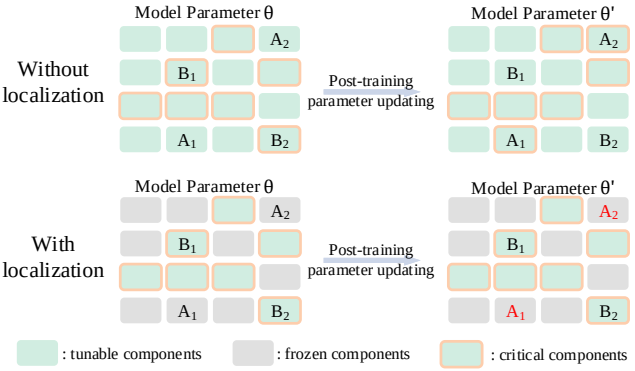


Figure 1: Differences in mechanism localization in SFT with and without localization

- **Divergent Free Evolution:** Without localization, critical components evolve freely during SFT, exhibiting distinct structural patterns: attention mechanisms undergo drastic shifts, whereas MLP components remain relatively stable.
- **Temporal Lag of Static Localization:** Because circuits inherently exhibit free evolution, utilizing current parameter states as a reference for future parameters suffers from severe latency and temporal lag.
- **The Illusion of Effectiveness:** The perceived success of existing Mechanistic Localization methods relies heavily on their application to MLP-dominated, knowledge-centric downstream tasks(e.g., knowledge editing, unlearning).

Ultimately, this paper reveals that Mechanistic Localization exhibits a critical **lag** during dynamic updates. To better optimize target performance and minimize mechanistic conflicts, we explore the necessity of more advanced, dynamic localization paradigms in Section 6.

[3]Specifically, independent parameter matrices like W_q, W_k, W_v, W_o in attention heads and $W_{\text{up}}, W_{\text{down}}$ in MLPs.

2 Preliminaries

We denote a well-trained LLM as $\mathcal{M} = \langle \mathcal{G}, \theta \rangle$, where θ represents the states of all trainable parameters. The computational graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ models the forward pass, with $\mathcal{V}$ comprising all components (i.e., parameter matrices such as $W_q, W_k, W_v, W_o, W_{\text{up}}, W_{\text{down}}$) and $\mathcal{E}$ denoting their activation connections (e.g., $W_o \to W_{\text{up}}$).

2.1 Circuit Discovery

We employ circuit discovery as our foundational mechanistic interpretability technique. Compared to alternatives, it better deconfounds correlations (vs. gradient/magnitude methods [11, 12]), captures holistic mechanisms (vs. causal interventions [13]), and provides stronger theoretical foundations (vs. probing [14] or vocabulary lenses [15]). Given a dataset $\mathcal{D}$, circuit discovery seeks to identify a minimal subgraph (circuit) $\mathcal{C} = \langle \mathcal{V}_\mathcal{C}, \mathcal{E}_\mathcal{C} \rangle \subset \mathcal{G}$ that faithfully captures the task-relevant behavior of $\mathcal{D}$ [16, 9, 17], optimized via:

$$\arg\min_{\mathcal{C}} \mathbb{E}_{x \in \mathcal{D}}[D(p_\mathcal{G}(y|x)||p_\mathcal{C}(y|x))], \quad \text{s.t.} \quad 1 - |\mathcal{C}|/|\mathcal{G}| \geq s \tag{1}$$

where s is the predefined sparsity constraint and D measures the output divergence between the full graph $\mathcal{G}$ and the circuit $\mathcal{C}$. Components (nodes and edges) in $\mathcal{C}$ whose causal contribution exceeds a predefined significance threshold τ are identified as the **critical components** responsible for executing task $\mathcal{D}$.

2.2 Post-Training SFT Objective

In post-training SFT, the primary goal is to adapt the model to a new **target task** with task dataset $\mathcal{D}_t$ while preserving pre-existing capabilities on general tasks (hereafter referred to as **pervasiveness tasks**, with datasets $\mathcal{D}_p$) [6]. Given initial parameters θ, target dataset $\mathcal{D}_t$ with inputs x and target outputs y_t, and pervasiveness dataset $\mathcal{D}_p$ with output distribution $p(y|x, \theta)$, the SFT optimization objective is naturally formulated as:

$$\min_{\theta'} \mathbb{E}_{(x,y_t) \in \mathcal{D}_t}[\mathcal{L}(y_t|x; \theta')] + \lambda \mathbb{E}_{(x,y) \in \mathcal{D}_p}[\mathcal{L}(y|x; \theta')] \tag{2}$$

where $\lambda \geq 0$ balances target adaptation and pervasiveness capability retention, and θ' represents the tuned parameter state.

2.3 Locate-then-Finetune Paradigm

Building upon circuit discovery and SFT, the **Locate-then-Finetune** paradigm executes in two sequential phases:

1. Locating Phase (Mechanistic Localization): Given pre-tuning parameters θ, we first apply circuit discovery (Eq. 1) on $\mathcal{D}_t$ and $\mathcal{D}_p$ to extract target circuit $\mathcal{C}_t = \langle \mathcal{V}_t, \mathcal{E}_t \rangle$ and pervasiveness circuit $\mathcal{C}_p = \langle \mathcal{V}_p, \mathcal{E}_p \rangle$, respectively. To disentangle target-specific components while preventing degradation on pervasiveness tasks, a Boolean solver $\mathcal{F}_{\text{logic}}$ [18] partitions the full component space $\mathcal{V}$ into tunable components $\mathcal{V}_{\text{tuned}}$ and frozen components $\mathcal{V}_{\text{freeze}}$:

$$\mathcal{V}_{\text{tuned}}, \mathcal{V}_{\text{freeze}} = \mathcal{F}_{\text{logic}}(\mathcal{G}, \mathcal{C}_t, \mathcal{C}_p), \quad \text{s.t.} \quad \mathcal{V}_{\text{tuned}} \cup \mathcal{V}_{\text{freeze}} = \mathcal{V}, \quad \mathcal{V}_{\text{tuned}} \cap \mathcal{V}_{\text{freeze}} = \emptyset \tag{3}$$

Accordingly, model parameters are partitioned into trainable subset θ_{tuned} (corresponding to $\mathcal{V}_{\text{tuned}}$) and frozen subset θ_{freeze} (corresponding to $\mathcal{V}_{\text{freeze}}$).

2. Finetuning Phase (SFT): During fine-tuning, θ_{freeze} remains strictly fixed at its initial state. Gradient updates are exclusively applied to θ_{tuned} to minimize the SFT loss (Eq. 2), yielding the updated parameters θ': $\theta' = \theta'_{\text{tuned}} \cup \theta_{\text{freeze}}$,

Recent literature has advanced both phases: localization enhancements include multi-task joint attribution [19, 2] and low-dimensional projections [20]; meanwhile, fine-tuning enhancements incorporate diverse optimization objectives such as negative preference optimization [21, 22, 23].

3 Rethinking the Static Assumption: Pitfalls and Observational Strategy

While the Locate-then-Finetune paradigm has achieved empirical popularity, its fundamental validity hinges on a crucial, yet largely unverified, static assumption. In this section, we formally articulate this core assumption, highlight its potential theoretical flaws, and outline our circuit-level observational framework designed to systematically test it.

3.1 Core Assumption and Its Potential Pitfalls

The Locate-then-Finetune paradigm operates on the fundamental premise that the pre-tuning parameter state θ provides an accurate, static roadmap for future dynamic parameter updates. Formally, let $\mathcal{V}'_{\text{critical}}$ represent the genuinely critical components required to execute the target task $\mathcal{D}_t$ at the updated parameter state θ'. The prevailing literature implicitly posits the following **Static Inclusion Assumption**:

$$\mathcal{V}'_{\text{critical}} \subseteq \mathcal{V}_{\text{tuned}}, \quad \text{such that} \quad \mathcal{V}'_{\text{critical}} \cap \mathcal{V}_{\text{freeze}} = \emptyset \tag{4}$$

In other words, existing methods assume that all neural components necessary for future task adaptation are fully encompassed within the pre-tuning localized region $\mathcal{V}_{\text{tuned}}$, and will never drift or migrate into the frozen region $\mathcal{V}_{\text{freeze}}$ during iterative fine-tuning.

However, this assumption is inherently vulnerable under common fine-tuning conditions. Consider a scenario where the pre-tuning model does not possess mastery over $\mathcal{D}_t$ at all (e.g., zero initial accuracy). In such cases, mechanistic localization performed on θ can not isolate any components associated with the task. It cannot anticipate where the optimal representations will settle once the model genuinely learns the task. Prematurely freezing $\mathcal{V}_{\text{freeze}}$ forcefully restricts optimization to a wrong subnetwork, potentially impeding the natural evolutionary trajectory of the target mechanism.

3.2 Proposed Circuit-Level Observational Framework

To rigorously evaluate whether Assumption (Eq. 4) holds, we translate these conceptual concerns into three concrete, observable circuit properties that explicitly bridge the static inclusion assumption with dynamic SFT behaviors:

- **Neural Migration:** If the assumption is violated, critical pathways migrate beyond $\mathcal{V}_{\text{tuned}}$ into the frozen region $\mathcal{V}_{\text{freeze}}$, which manifests as measurable component-level structural transfers across the computational circuit graph.
- **Semantic Stability:** If the model genuinely masters the target task during SFT, the underlying task mechanism should consolidate into a robust circuit.
- **Cross-Task Conflict:** If $\mathcal{V}'_{\text{critical}} \subset \mathcal{V}_{\text{tuned}}$; mechanistically proven by [18], this manifests as a decreasing degree of Boolean logical conflict between target circuit $\mathcal{C}_t$ and pervasiveness circuit $\mathcal{C}_p$.

Observational Pipeline: As illustrated in our experimental design, we systematically track and extract logical circuits $\mathcal{C}_t^{(s)}$ and $\mathcal{C}_p^{(s)}$ across intermediate fine-tuning checkpoints $s \in \{0, 1, \ldots, S\}$. We define three novel metrics—*Circuit Distance* (CD), *Circuit Stability* (CS), and *Circuit Conflict* (CC)—to quantify these properties (detailed in Section 4).

Crucially, to isolate the structural impact of parameter freezing itself, we compare two parallel evolution regimes: **Free Evolution** (unconstrained full-parameter SFT) and **Localization Evolution** (SFT with localized parameter freezing). This comparative setup enables us to ablate the inductive bias introduced by static localization and directly observe how task mechanisms naturally evolve versus how they behave under constrained fine-tuning.

4 Evaluation Metrics

This section introduces three metrics (summarized in Table 1) to evaluate mechanism evolution. To address **RQ1**, **Circuit Distance** (CD) and **Circuit Stability** (CS) evaluate a single mechanism's evolution: given a target task's logical circuit (Appendix A), CD measures component migration, while CS assesses knowledge consolidation. To address **RQ2**, we propose **Circuit Conflict** (CC) to

quantify inter-mechanism interference. Combined with intrinsic task performance metrics - which gauge capability enhancement or retention - CC comprehensively evaluates multi-task interactions within the Locate-then-Finetune paradigm.

Table 1: An overview of the three circuit metrics.

Metrics	Object	Data Format	Function
CD	Single circuit	Circuit Graph	How do components transfer?
CS	Single circuit	Circuit Distribution	How well is the mechanism learned?
CC	Multiple circuits	CNF from Circuits	How do mechanisms conflict?

4.1 Circuit Distance (CD)

In circuit discovery, an edge $W_i \to W_j \in \mathcal{C}$ is identified by measuring the output variance under causal interventions. An edge is retained if its causal effect $I(W_i \to W_j)$ exceeds a threshold τ:

$$I(W_i \to W_j) = |\mathbb{L}(x|\text{do}(W_i \to W_j)) - \mathbb{L}(x)| > \tau \tag{5}$$

where $\mathbb{L}$ denotes the output logits, and $\text{do}(\cdot)$ represents activation patching. For these interventions, we default to interchange ablation [24, 25, 26, 27].

To quantify the migration of critical pathways across parameter states θ and θ', we employ the Manhattan distance over the computational graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$. Specifically, for a computational graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ with N components, we associate each edge in $\mathcal{E}$ with its causal effect, denoted as $\mathcal{E} = \{W_i \to W_j, I(W_i \to W_j)\}$. This approach captures continuous variations more effectively than discrete metrics (e.g., Hamming distance). The Circuit Distance (CD) is defined as:

$$CD = D_{\mathcal{G}}(\mathcal{G}^{\theta}, \mathcal{G}^{\theta'}) = \sum_{(W_i, W_j \in \mathcal{E})} \left| I(W_i \to W_j)^{\theta} - I(W_i \to W_j)^{\theta'} \right| \tag{6}$$

To account for varying logit baselines across tasks, we normalize CD using the maximum empirical range of $I(\cdot)$. Ultimately, CD reflects the extent of mechanism transformation by aggregating the absolute shifts in causal effects across all components.

4.2 Circuit Stability (CS)

Beyond tracking mechanism migration, it is crucial to assess the model's mastery of the mechanism. Recent studies [28] indicate that if the learning of the target mechanism has not converged, different sampled subsets of $\mathcal{D}_t$ will yield significantly divergent circuits. We quantify this using **Circuit Stability** (CS), defined as the expected Spearman's rank correlation (ρ) between circuits derived from two i.i.d. subsets s, s' (each 20% of $\mathcal{D}_t$):

$$CS(\theta') = \mathbb{E}_{s,s' \in \mathcal{D}_t}[\rho(\mathcal{C}_s^{\theta'}, \mathcal{C}_{s'}^{\theta'})] \tag{7}$$

Here, ρ ranks the causal effects of all edges. Specifically, we take circuits $\mathcal{C}_s$ and $\mathcal{C}_{s'}$, and rank $\mathcal{E} = \{W_i \to W_j, I(W_i \to W_j)\}_{(i,j)}^{N}$ based on $\mathcal{C}_s(\mathcal{E})$ and $\mathcal{C}_{s'}(\mathcal{E})$, respectively. Therefore, CS reflects the stability trend of the target mechanism across two parameter states by measuring the correlation between circuits derived from different randomly sampled subsets. A large $|CS|$ indicates high cross-sample consistency, meaning knowledge is effectively consolidated. Conversely, a small $|CS|$ implies the information remains scattered and unsettled, undergoing further refinement. Combining CD and CS yields four joint states:

- **Large CD, Large $|CS|$**: The model preserves the firmly consolidated information of the mechanism but actively migrates its storage to a new set of components.
- **Small CD, Small $|CS|$**: The model continues to utilize the original components but actively updates and refines the unsettled internal information within them.
- **Large CD, Small $|CS|$**: The model undergoes a reorganization, simultaneously updating the unsettled internal information and migrating its storage to a new set of components.
- **Small CD, Large $|CS|$**: The model maintains a highly stable state, retaining both the consolidated information of the mechanism and the original components storing it.

An intuitive, albeit imprecise, metaphor is that CD determines whether to change the *glass* holding the cocktail, while CS determines whether to refine the cocktail's *recipe*. CD and CS not only reflect the evolutionary state of a single mechanism but also serve as a "baseline evolution" for studying multiple mechanisms. By comparing against this baseline, we can better analyze the practical impact of localization on multiple mechanisms.

4.3 Circuit Conflict (CC)

We define **conflicting components** as critical components shared across multiple optimization targets' circuits. Reducing these components typically enhances post-training performance [18], as sharing induces optimization competition rather than synergy. For instance, a polysemantic component W_i encoding both "sports" and "geography" faces competing gradient requirements during multi-objective optimization within "sports" and "geography" tasks, preventing simultaneous optimality. Conversely, disentangling these semantics into distinct components W_m and W_n enables independent tuning to task-specific peaks. Therefore, a decreasing trend in conflicting components during evolution is highly desirable, signifying a shift toward specialized neural representations that facilitate superior independent performance.

However, identifying conflicting components by simply intersecting circuits yields biased conclusions due to component non-exclusivity (e.g., redundant components effectively forming an "OR gate"). To resolve this, we leverage logical circuits [18] to map task circuits into Conjunctive Normal Form (CNF) clauses, framing conflict detection as a Boolean satisfiability (SAT) problem. Let Φ_t and Φ_p denote the CNFs for the target and pervasiveness circuits, respectively. We define Circuit Conflict (CC) as:

$$CC(\theta) = \min_{n} \text{UNSAT}(\Phi_t^\theta \wedge \Phi_p^\theta) \tag{8}$$

where UNSAT computes the **UNSAT Core**—the minimum number of conflicting clauses—using a SAT solver [29, 30, 31]. This formulation naturally expands for multiple pervasiveness tasks (e.g., $\Phi_t \wedge \Phi_{p1} \wedge \dots$).

By extracting the UNSAT Core across parameter states, CC strictly quantifies the evolutionary trend of conflicts. A larger CC signifies intensified interference from pervasiveness tasks, which inherently hinders optimal target performance. Ultimately, CC bypasses superficial performance fluctuations, directly revealing whether multiple mechanisms evolve synergistically.

5 Experiments

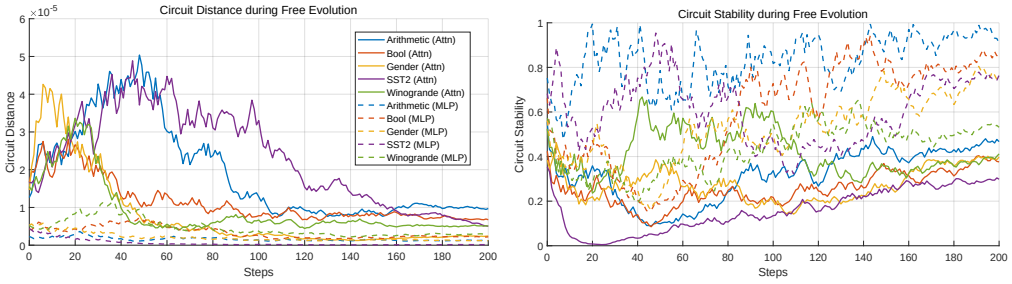
Our evaluation framework is grounded in a SFT pipeline. The primary optimization objective is to master a target task while preserving performance on a pervasiveness task. We analyze the underlying mechanistic dynamics by evaluating task circuits at each SFT step (extraction details in Appendix A).

Building upon this, we structure our experiments around two distinct SFT pipelines:

- **Free Evolution:** Without Mechanistic Localization, we fine-tune all parameters to investigate whether, how, to what extent, and why circuits change during parameter evolution.
- **Localization Evolution:** Adhering to the "Locate-then-Finetune" paradigm (Section 2.3), we identify critical components (Appendix A), freeze the others, and perform SFT. By comparing this against free evolution, we explore whether localization genuinely benefits parameter updates and investigate why existing methods ostensibly improve task performance.

We employ Mistral-7B and LLaMA3-8B[4] as baseline models. For target and pervasiveness tasks, we select 15 datasets widely adopted in mechanistic interpretability and post-training research: OpenBookQA [32], Gender [33], RTE [34], IOI [35], Docstring [36], SST2 [37], Winogrande [38], Reverse [39], Greater Than [40], FEVER [41], zsRE [42], Induction [9], Bool [43], Arithmetic [44], and SA [45]. We track SFT evolution across 10 epochs, subdividing each into 20 observation steps to capture granular dynamics. Comprehensive setups are detailed in Appendix B. The performance fluctuations of all SFT experiments across 5 random seeds are tightly bound within a narrow 95% confidence interval (t=2.776) and we report the representative outcomes for clarity.

[4]Mistral-7B: `https://huggingface.co/mistralai/Mistral-7B-v0.1`. Llama3-8B: `https://huggingface.co/meta-llama/Meta-Llama-3-8B`



(a) Circuit Distance (b) Circuit Stability

Figure 2: line plots of different target tasks on the Mistral-7B model in terms of Circuit Distance (CD) and Circuit Stability (CS).

5.1 Free Evolution

In this subsection, we focus on **RQ1** (*Without localization, do the critical components of the target skill change during parameter updates? If so, how do they evolve?*).

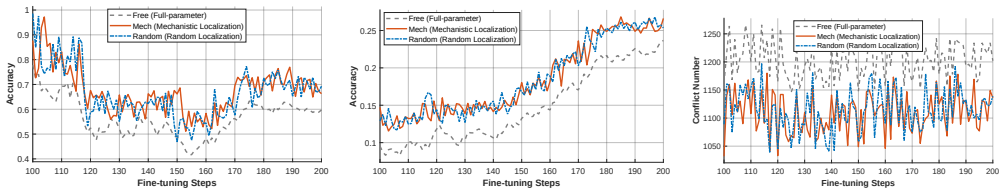
First, we evaluate mechanism evolution on Mistral-7B and LLaMA3-8B across five target tasks (Arithmetic, SST2, Winogrande, Bool, Gender) against a general pervasiveness task (OpenBookQA). Circuits are extracted via the Logical Circuit Framework (Appendix A.3). Figure 2 plots the CD and CS trajectories during SFT on Mistral-7B, where CD tracks the shift from the initial parameters (θ^0) to the current step (θ^s). Full results, including LLaMA3-8B findings and additional task performance metrics, are detailed in Appendix C. Without localization, task mechanisms spontaneously shift during parameter updates, driving substantial circuit migration. Notably, even well-mastered tasks like Gender (initial accuracy $> 85\%$) undergo drastic migration, replacing numerous critical components (Table 9). Ablation studies (Tables 9 and 6) confirm this significant transfer persists even without the pervasiveness task and at 100% initial accuracy. Ultimately, this dynamic nature confirms that static mechanistic conclusions (circuits) derived from current parameters cannot reliably represent future neural contributions. Moreover, our conclusions can also explain some existing findings [46, 47, 48, 49] regarding changes in the SFT mechanism. The detailed analysis are shown in Appendix D.

Second, we track the evolution of attention (W_q, W_k, W_v, W_o; denoted *Attn*) and MLP ($W_{\text{up}}, W_{\text{down}}$; denoted *MLP*) components. As shown in Figure 2, *Attn* components exhibit significantly higher CD than *MLP* components throughout SFT. Prior mechanistic studies [3, 50] associate Attention with inter-token "skills" (e.g., induction heads dedicated to processing "A B ... A" patterns, or syntax heads handling structures like "The + Noun") and MLPs with semantic "knowledge" (e.g., factual concepts). Combining these insights reveals a fundamental evolutionary pattern: "skill"-centric *Attn* circuits are highly volatile and prone to migration during updates, whereas "knowledge"-centric *MLP* circuits remain largely inert. This dichotomy is further corroborated by CS trajectories, where *MLP* stability substantially exceeds *Attn* stability.

Finally, we designed a series of ablation experiments to observe the impact of varying factors on Circuit Distance (CD) and Circuit Stability (CS). Ultimately, we identified five key factors that significantly influence circuit evolution: **Task Type** ("Skill" task and "Knowledge" task), **Degree of Pervasiveness**, **Dataset Size**, **Conflict Proportion**, **Initial Mastery**. Table 2 summarizes these factor-metric correlations, with comprehensive ablation in Appendix C.

Table 2: Summary of key factors on free evolution

Factor	CD **of Attn**	CD **of MLP**	CS **of Attn**	CS **of MLP**
"Skill" tasks	↑	-	↓	-
"Knowledge" tasks	-	↓	-	↑
Pervasiveness ↑	↑	↑	-	-
Dataset Size ↑	-	-	↑	↑
Conflicting↑	↑	↑	↓	↓
Mastered tasks	↓	↓	↑	↑
Unmastered tasks	↑	↑	↓	↓



(a) Target Accuracy (b) Pervasiveness Accuracy (c) Circuit Conflict

Figure 3: Target Task Accuracy, Pervasiveness Task Accuracy, and Circuit Conflict of Arithmetic Task with localization.

5.2 Localization Evolution

In this section, we transition to evaluating the dynamics of **Localization Evolution**. Maintaining the identical experimental setup as in Section 5.1, we compare 3 localization strategies during SFT:

- **Free (Baseline):** Full-parameter SFT without freezing any parameters.
- **Mech (Mechanistic Localization):** Localizing critical components based on the initial parameter state (θ) via mechanistic interpretability, while freezing all non-critical components.
- **Random:** Randomly updating the exact same number of components as identified in the *Mech* strategy (same ratio between Attn and MLP), while freezing the remainder.

Through the comparative analysis of these three strategies, we aim to systematically address **RQ2** (*Does post-training with localization truly improve performance on the target task and mitigate conflicts with pervasiveness tasks?*).

5.2.1 The Contribution of Static Localization

To address **RQ2**, we compared three localization strategies (**Free**, **Mech**, and **Random**) on Mistral-7B. Consistent with previous settings, we paired five target tasks (Arithmetic, Bool, Gender, SST2, Winogrande) with the OpenBookQA pervasiveness task. For the *Mech* strategy, we isolated 800 critical components (the predetermined circuit scale). To ensure a rigorously controlled comparison, the *Random* strategy correspondingly updated exactly 800 randomly selected components.

Figure 3 illustrates the results for the Arithmetic task across three metrics: Target Task Accuracy, Pervasiveness Task Accuracy, and Circuit Conflict (The results of the other 4 tasks are shown in Appendix E.1.). A compelling observation emerges: First, the Target Task Accuracy fails to surpass its pre-SFT baseline, a phenomenon that starkly contrasts with typical outcomes observed in single-objective optimization. Coupled with the evidence of circuit conflicts presented in Figure 3 (c), this substantiates that conflicts arising from multi-task optimization inevitably degrade individual task performance. Furthermore, a compelling observation emerges: while the *Mech* strategy significantly outperforms the *Free* baseline in terms of accuracy and successfully maintains a lower conflict level, it exhibits no substantial advantage over the *Random* strategy. This outcome reveals that while localization *in general* does benefit task performance, this improvement ostensibly does not stem from the mechanistic interpretability guidance of the circuit.

The CD and CS metrics provided in Appendix E.2 further corroborate this: compared to free evolution, localization induces greater Circuit Distance and variations in Circuit Stability. In other words, localization paradoxically renders the circuit's evolution more uncontrollable and unstable, and the improvement in task performance may stem from the reduction in parameters, which makes task optimization simpler. This validates our preceding hypothesis: Mechanistic Localization based on the current parameter state identifies only a *fraction* of the components that genuinely contribute to future parameter updates. Because the **unidentified critical components** are prematurely **frozen**, they impede normal evolutionary dynamics, rendering circuit evolution substantially more difficult. Consequently, from a macroscopic perspective, its performance is virtually indistinguishable from "random localization."Furthermore, Appendix E.3 provides additional validation for this hypothesis. Expanding the circuit scale—thereby encompassing more "**unidentified critical components**"—allows Mechanistic Localization to surpass Random Localization.

5.2.2 Why existing methods show significant improvements?

However, prevalent downstream applications (e.g., LLM unlearning, Knowledge Editing) report Mechanistic Localization vastly outperforming Random Localization, starkly contradicting **RQ2**. We hypothesize this discrepancy arises because these applications predominantly target MLP-governed, knowledge-centric tasks. As established in **RQ1** (Section 5.1), MLP circuits inherently resist component migration. Consequently, the divergence between current and updated circuits is significantly diminished. This minimal evolutionary drift creates an "illusion of effectiveness"—the false premise that the "current circuit" reliably guides the "future circuit."

To empirically validate this hypothesis, we conducted a comparative analysis within the LLM unlearning paradigm, contrasting mainstream unlearning on the WMDP-Bio dataset against unlearning the Induction task. The WMDP-Bio dataset comprises biological knowledge and related factual information; hence, unlearning WMDP-Bio is a quintessential knowledge-centric task dominated by MLP circuits. Conversely, the Induction task relies on a skill-centric circuit predominantly driven by Attention mechanisms, making its unlearning an Attention-dominated process. We evaluated these two target tasks with OpenBookQA acting as the pervasiveness task. The comprehensive metrics tracked include "FE" (Forget efficacy) is measured as 1-accuracy on unlearning task and "RU" (Retain utility) is measured as accuracy of pervasiveness task, Circuit Distance of Attention (CD_{Attn}), Circuit Distance of MLP (CD_{MLP}), Circuit Stability of Attention (CS_{Attn}), Circuit Stability of MLP (CS_{MLP}), and Circuit Conflict (CC).

Table 3: Performance of WMDP-Bio and Induction tasks on LLM unlearning

Datasets	Localization	FE	RU	CD_{Attn}	CD_{MLP}	CS_{Attn}	CS_{MLP}	CC
WMDP-Bio	Free	0.685	0.467	0.8412	0.5749	0.7819	0.9557	1055
	Random	0.716	0.441	1.1243	0.9557	0.5219	0.8396	1242
	Mech	0.736	0.457	1.0519	0.7391	0.6467	0.9082	1091
Induction	Free	0.873	0.324	1.2479	0.8949	0.7544	0.8846	1233
	Random	0.952	0.359	1.5866	1.0547	0.8137	0.8573	1153
	Mech	0.955	0.367	1.5593	1.1134	0.7968	0.8682	1129

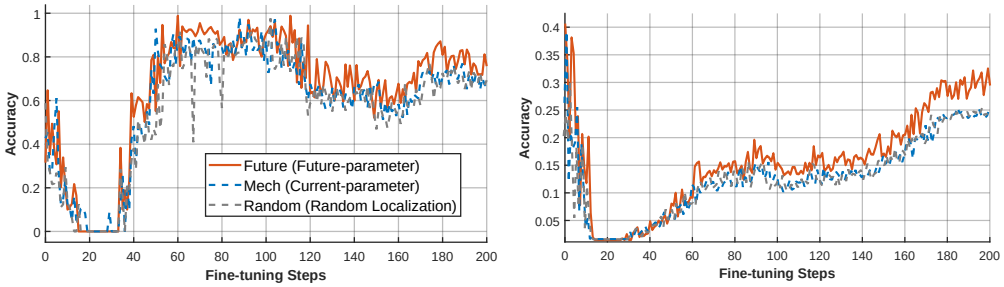
Table 3 corroborates our hypothesis: because the circuits of knowledge-centric tasks are largely composed of MLPs and their associated edges, the immense parameter volume and dense knowledge superposition within MLPs render them highly resistant to neural migration. In stark contrast, skill-centric tasks governed by Attention mechanisms are much more susceptible to interference due to their smaller parameter footprint, leading to a substantially higher proportion of component migration. In Appendix F, we provide supplementary sampling results during the optimization process when these two tasks are treated as independent optimization objectives, further substantiating this conclusion.

6 Discussion: The Path to Predictive Localization

Section 5 demonstrates that task mechanisms inherently undergo "free evolution." Due to these dynamic shifts, static interpretability conclusions suffer from severe temporal latency, fundamentally failing to guide future parameter updates. To make Mechanistic Localization practical, we must endow it with "foresight"—the ability to predict future mechanisms.

Ideally, successfully forecasting the post-update circuit from the current parameter state would enable flawless critical component localization. Accordingly, we conducted a preliminary exploration using the circuit obtained *after* free evolution SFT as a surrogate for the current circuit, introducing a paradigm we term *Future Mechanistic* localization. Figure 4 shows that this approach yields substantial advantages in both Target (T-Acc) and Pervasiveness (P-Acc) Accuracy on the Arithmetic task (additional metrics in Appendix G), powerfully corroborating our hypothesis and charting a clear trajectory for future research.

Furthermore, Appendix G presents a comprehensive analysis of contemporary Mechanistic Localization methodologies. Moving beyond strictly circuit-centric approaches, we evaluate two other predominant paradigms: gradient-based methods and intervention-based methods. Our empirical results reveal that critical components localized via gradient-based techniques exhibit markedly superior performance in terms of Circuit Distance (CD). This suggests a compelling insight: leveraging



(a) Task Accuracy (b) Pervasiveness Accuracy

Figure 4: Line plots of Future-Localization.

current gradient signals to approximate or extrapolate the mechanistic structures of future parameter states represents a highly promising frontier for the evolution of predictive Mechanistic Localization.

7 Conclusion

In this paper, we explore whether interpretability conclusions derived from current parameter states offer predictive guidance for future parameter updates. To investigate this, we constructed a "Locate-then-Finetune" pipeline using circuit discovery for Mechanistic Localization. By tracking circuits across SFT steps via three novel metrics—Circuit Distance (CD), Circuit Stability (CS), and Circuit Conflict (CC)—we systematically evaluated the evolutionary dynamics of task mechanisms. Extensive experiments substantiate three core conclusions:

- **Divergent Free Evolution:** Without localization, circuits naturally undergo "free evolution." Attention-driven skill tasks experience drastic structural shifts, whereas MLP-reliant knowledge tasks evolve much more gradually.
- **Temporal Lag of Static Localization:** Due to this inherent free evolution, utilizing current parameter states as a reference suffers from severe temporal latency. Consequently, static Mechanistic Localization fails to reliably guide future dynamic updates.
- **The Illusion of Effectiveness:** The perceived success of existing localization methods stems from their evaluation on MLP-dominated tasks. The inherently slower evolution of these tasks coincidentally masks the latency of static circuits, creating a false sense of efficacy.

Finally, we discuss potential research directions for achieving effective Mechanistic Localization and outline the limitations of this work in Appendix H.

Acknowledgments and Disclosure of Funding

This research / project is supported by the National Research Foundation, Singapore under its National Large Language Models Funding Initiative, (AISG Award No: AISG-NMLP-2024-005), and under its National Cybersecurity R&D Programme (CRPO-GC5-NTU-011). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore. Moreover, it is also supported by the Xi'an Jiaotong University - China Mobile Communications Group Co., Ltd. Joint Institute.

References

[1] Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. Depn: Detecting and editing privacy neurons in pretrained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2875–2886, 2023.

[2] Yongce Li, Chung-En Sun, and Tsui-Wei Weng. Effective skill unlearning through intervention and abstention. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6358–6371, 2025.

[3] 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.

[4] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, 2022.

[5] Lecheng Yan, Ruizhe Li, Guanhua Chen, Qing Li, Jiahui Geng, Wenxi Li, Vincent Wang, and Chris Lee. Spurious rewards paradox: Mechanistically understanding how rlvr activates memorization shortcuts in llms. *arXiv preprint arXiv:2601.11061*, 2026.

[6] Hengyuan Zhang, Zhihao Zhang, Mingyang Wang, Zunhai Su, Yiwei Wang, Qianli Wang, Shuzhou Yuan, Ercong Nie, Xufeng Duan, Qibo Xue, et al. Locate, steer, and improve: A practical survey of actionable mechanistic interpretability in large language models. *arXiv preprint arXiv:2601.14004*, 2026.

[7] 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, editors, *Advances in Neural Information Processing Systems*, volume 38, pages 150511–150540. Curran Associates, Inc., 2025.

[8] Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *Advances in Neural Information Processing Systems*, 36:17643–17668, 2023.

[9] 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.

[10] 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.

[11] 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*, pages 681–691, 2016.

[12] 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)*, pages 5701–5715, 2024.

[13] Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7035–7052, 2023.

[14] Tianjie Ju, Weiwei Sun, Wei Du, Xinwei Yuan, Zhaochun Ren, and Gongshen Liu. How large language models encode context knowledge? a layer-wise probing study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8235–8246, 2024.

[15] Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

[16] 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.

[17] Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*, 2024.

[18] 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*, 2026.

[19] Jinghan Jia, Jiancheng Liu, Yihua Zhang, Parikshit Ram, Nathalie Baracaldo, and Sijia Liu. Wagle: Strategic weight attribution for effective and modular unlearning in large language models. *Advances in Neural Information Processing Systems*, 37:55620–55646, 2024.

[20] Aashiq Muhamed, Jacopo Bonato, Mona T Diab, and Virginia Smith. Saes can improve unlearning: Dynamic sparse autoencoder guardrails for precision unlearning in llms. In *Second Conference on Language Modeling*, 2025.

[21] Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*, 2024.

[22] Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pages 243–254. PMLR, 2022.

[23] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. In *First Conference on Language Modeling*, 2024.

[24] Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*, 2024.

[25] 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.

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

[27] Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching. *arXiv preprint arXiv:2304.05969*, 2023.

[28] Alan Sun. Circuit stability characterizes language model generalization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9025–9040, 2025.

[29] Daniel Selsam and Nikolaj Bjørner. Guiding high-performance sat solvers with unsat-core predictions. In *International conference on theory and applications of satisfiability testing*, pages 336–353. Springer, 2019.

[30] Alessandro Cimatti, Alberto Griggio, and Roberto Sebastiani. A simple and flexible way of computing small unsatisfiable cores in sat modulo theories. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 334–339. Springer, 2007.

[31] Nicolás D'Ippolito, Marcelo F Frias, Juan P Galeotti, Esteban Lanzarotti, and Sergio Mera. Alloy+ hotcore: A fast approximation to unsat core. In *International Conference on Abstract State Machines, Alloy, B and Z*, pages 160–173. Springer, 2010.

[32] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

[33] Chris Mathwin, Guillaume Corlouer, Esben Kran, Fazl Barez, and Neel Nanda. Identifying a preliminary circuit for predicting gendered pronouns in gpt-2 small. *URL: https://itch. io/jam/mechint/rate/1889871*, page 2, 2023.

[34] Ido Dagan, Dan Roth, Fabio Zanzotto, and Mark Sammons. *Recognizing textual entailment: Models and applications*. Springer Nature, 2022.

[35] 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*.

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

[37] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[38] Winogrande: An adversarial winograd schema challenge at scale. 2019.

[39] David Lindner, János Kramár, Sebastian Farquhar, Matthew Rahtz, Tom McGrath, and Vladimir Mikulik. Tracr: Compiled transformers as a laboratory for interpretability. *Advances in Neural Information Processing Systems*, 36:37876–37899, 2023.

[40] 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.

[41] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[42] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In Roger Levy and Lucia Specia, editors, *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[43] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, 2023.

[44] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. 2020.

[45] Lei Yu, Jingcheng Niu, Zining Zhu, and Gerald Penn. Functional faithfulness in the wild: Circuit discovery with differentiable computation graph pruning. *arXiv preprint arXiv:2407.03779*, 2024.

[46] Vishnu Kabir Chhabra, Ding Zhu, and Mohammad Mahdi Khalili. Neuroplasticity and corruption in model mechanisms: A case study of indirect object identification. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3099–3122, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[47] Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P. Dick, Hidenori Tanaka, Edward Grefenstette, Tim Rocktäschel, and David Krueger. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*, 2024.

[48] Nikhil Prakash, Tamar Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 8057–8082, 2024.

[49] Xu Wang, Yan Hu, Wenyu Du, Reynold Cheng, Benyou Wang, and Difan Zou. Towards understanding fine-tuning mechanisms of LLMs via circuit analysis. In *Forty-second International Conference on Machine Learning*, 2025.

[50] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, 2021.

[51] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[52] 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.

[53] 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.

[54] Vaidehi Patil, Peter Hase, and Mohit Bansal. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. In *The Twelfth International Conference on Learning Representations*, 2023.

A Details of Logical Circuit Framework

At first, we systematically introduce three fundamental circuit logic types: the **AND** gate, **OR** gate, and **ADDER** gate [7].

Definition 1. *We assume a common paradigm in which a receiver node B, which is connected by more than 1 sender node $A_1, A_2, ...$ For any edge $A_i \to B$, we use binary values '0' and '1' to represent the activation state of a node. Specifically, $A_i = 0$ indicates that node A_i is removed, ablated, or deactivated, whereas $A_i = 1$ indicates that node A_i is retained and active. When the sender nodes are ablated, the effect of node B on the output exhibits three distinct patterns, which are as follows:*

***AND**: All sender nodes satisfy an AND logical relationship with the receiver node, i.e., $B = A_1 \wedge A_2 \wedge \ldots$. In this case, node B exerts a significant effect on the output only if all of its sender nodes are retained. If even a single sender node is ablated, the effect of B on the output is nearly eliminated.*

***OR** gate: All sender nodes satisfy an OR logical relationship with the receiver node, i.e., $B = A_1 \vee A_2 \vee \ldots$. In this case, node B always exerts a significant effect on the output if one or more of*

its sender nodes are retained. Only if all sender nodes are ablated, the effect of B on the output is nearly eliminated.

***ADDER** gate: all sender nodes satisfy an ADDER logical relationship with the receiver node, i.e., $B = A_1 + A_2 + \ldots$. In this case, node B exhibits its maximal effect on the output only when all of its sender nodes are retained. If any single sender node is ablated, the effect of B on the output is substantially diminished; when all sender nodes are ablated, B's effect on the output is reduced to zero. Accordingly, we define the state of B as taking values 0,1,2,., where the total number of distinct states equals the number of sender nodes.*

Theoretical analyses support the view that noising-based intervention is capable of recovering a complete AND gate but fails to recover a complete OR gate, whereas denoising-based intervention demonstrates the opposite pattern [24]. This asymmetry is straightforward to interpret. The noising-based intervention procedure corresponds to the transition from a clean activation state (state $= 1$) to a corrupted activation state (state $= 0$). Since all gates can be regarded as being initialized with activation states equal to 1, any transition to state $= 0$ induces a significant change in the effect of AND and ADDER gates on the output. Consequently, noising-based intervention can reliably identify AND and ADDER gates.

The **denoising-based intervention** first performs the corrupted run in the computational graph, and then replaces the corrupted activations with the clean activations. Those activations that lead to significant changes in the output ($\tilde{y}$) consist of the circuits. denoising-based intervention thus has the following objective:

$$\arg\min_{\mathcal{C}} \mathbb{E}_{(x,\tilde{x})\in\mathcal{T}}[D(p_{\mathcal{G}}(\tilde{y}|\tilde{x})||p_{\mathcal{C}}(\tilde{y}|\tilde{x},x))], \;\; s.t.\; 1-|\mathcal{C}|/|\mathcal{G}| \geq s \tag{9}$$

Conversely, the denoising-based intervention procedure corresponds to initialization with activation states equal to 0. In this case, any transition to state $= 1$ produces a significant change in the effect of OR and ADDER gates on the output.

Therefore, we denote the circuit constructed under the noising-based intervention strategy as $\mathcal{C}_{\text{Ns}}$, and the one constructed under the denoising-based intervention strategy as $\mathcal{C}_{\text{Dn}}$. Based on the above set-theoretic relationships between $\mathcal{C}_{\text{Ns}}$ and $\mathcal{C}_{\text{Dn}}$, we extract subsets of edges corresponding to AND, OR, and ADDER gates as follows:

- AND gate ($\mathcal{C}_{\text{AND}}$): edges that are present in $\mathcal{C}_{\text{Ns}}$ but absent from $\mathcal{C}_{\text{Dn}}$.
- OR gate ($\mathcal{C}_{\text{OR}}$): edges that are present in $\mathcal{C}_{\text{Dn}}$ but absent from $\mathcal{C}_{\text{Ns}}$.
- ADDER gate ($\mathcal{C}_{\text{ADDER}}$): edges that are shared between $\mathcal{C}_{\text{Ns}}$ and $\mathcal{C}_{\text{Dn}}$.

Therefore, we propose a combined **Ns+Dn** approach to recover logically complete gates. This method is compatible with a wide range of circuit discovery algorithms, introduces minimal additional computational overhead, and enables clear and effective separation of the three types of logic gates. Ns+Dn has the following objective:

$$\arg\min_{\mathcal{C}} \mathbb{E}_{(x,\tilde{x})\in\mathcal{T}}[D(p_{\mathcal{G}}(y|x)||p_{\mathcal{C}}(y|x,\tilde{x})) + D(p_{\mathcal{G}}(\tilde{y}|\tilde{x})||p_{\mathcal{C}}(\tilde{y}|\tilde{x},x))], \;\; s.t.\; 1-|\mathcal{C}|/|\mathcal{G}| \geq s \tag{10}$$

In the following sections, we provide a detailed exposition of the original design of each method under the Ns. strategy, the corresponding formulation under the Dn. strategy, and the final approach that integrates both—Ns.+Dn.—for recovering logically complete gates.

A.1 Greedy Search Example: ACDC

The ACDC method identifies important edges by iteratively removing each edge and observing the effect of this intervention on the model output. Edges whose removal causes an effect greater than a predefined threshold τ are retained, while those with an effect smaller than τ are pruned. The original algorithm (Ns. strategy), is outlined as follows:

In the **Ns.** strategy, $\mathcal{G}$ denotes the **clean run**, and $\mathcal{H} \setminus \{w \to v\}$ represents the replacement of the clean activation on the edge $w \to v$ with its corrupted activation. In contrast, under the **Dn.** strategy, $\mathcal{G}$ refers to the **corrupted run**, and $\mathcal{H} \setminus \{w \to v\}$ indicates the substitution of the corrupted activation on edge $w \to v$ with the corresponding clean activation.

Algorithm 1: The ACDC algorithm in Ns.

Data: Computational graph $\mathcal{G}$, dataset $(x_i)_{i=1}^n$, corrupted datapoints $(x'_i)_{i=1}^n$ and threshold $\tau > 0$.

Result: Subgraph $\mathcal{H} \subseteq \mathcal{G}$.

```
H ← G                                  // Initialize H to the full computational graph
H ← H.reverse_topological_sort()       // Sort H so output first
for v ∈ H do
    for w parent of v do
        H_new ← H \ {w → v}            // Temporarily remove candidate edge
        if D_KL(G||H_new) − D_KL(G||H) < τ then
            H ← H_new                  // Edge is unimportant, remove permanently
return H
```

In the combined **Ns.+Dn.** approach, the effects from both strategies are jointly considered. Specifically, the original pruning condition $D_{KL}(\mathcal{G} \,\|\, \mathcal{H}_{\text{new}}) - D_{KL}(\mathcal{G} \,\|\, \mathcal{H}) < \tau$ is replaced with the aggregated criterion: $D_{KL}(\mathcal{G}^{\text{clean}} \,\|\, \mathcal{H}_{\text{new}}) - D_{KL}(\mathcal{G}^{\text{clean}} \,\|\, \mathcal{H}) + D_{KL}(\mathcal{G}^{\text{corrupted}} \,\|\, \mathcal{H}_{\text{new}}) - D_{KL}(\mathcal{G}^{\text{corrupted}} \,\|\, \mathcal{H}) < \tau$.

A.2 Linear Estimation Example: EAP

The EAP method approximates the effect of each edge using the first-order term of its Fourier expansion, enabling the estimation of all edge effects with a single forward pass. It is important to note that, during the computation of each edge's effect, all other edges remain in their unpruned (active) state.

Specifically, Ns. has approximation:

$$L(x|do(\tilde{x}_i)) - L(x) \approx (\tilde{x}_i - x_i)^T \frac{\partial}{\partial x_i} L(x) \tag{11}$$

and Dn. has approximation:

$$L(\tilde{x}|do(x_i)) - L(\tilde{x}) \approx (\tilde{x}_i - x_i)^T \frac{\partial}{\partial \tilde{x}_i} L(\tilde{x}) \tag{12}$$

Therefore, the approximation for Ns.+Dn. is $(\tilde{x}_i - x_i)^T \frac{\partial}{\partial x_i} L(x) + (\tilde{x}_i - x_i)^T \frac{\partial}{\partial \tilde{x}_i} L(\tilde{x})$.

A.3 Differentiable Mask Example: EdgePruning

EdgePruning assigns a learnable mask to each node or edge, where the mask is reparameterized using the hard concrete distribution. In the Ns. setting, the optimization objective corresponds to Equation 1. Consequently, the objectives for the Dn. and Ns.+Dn. settings are given by Equation 9 and Equation 10, respectively.

In the Ns.+Dn. setting, directly optimizing both objectives jointly can lead to gradient interference and convergence to Pareto-optimal solutions, rather than a unified optimum. To address this, we independently compute the final mask values for Ns. and Dn. using Equations 1 and 9, and then obtain the mask for Ns.+Dn. by averaging the two.

Finally, we simplify the ADDER gate in the forget circuit to an OR gate, and the ADDER gate in the retain circuit to an AND gate.

B Experiment Details

The learning rate is grid-searched at 1×10^{-5} for each dataset. The parameter $\lambda = 1$, and we adopted AdamW [51] as the optimizer. All experiments were conducted on 16 NVIDIA RTX A100 GPUs.

We select a series of specific tasks as target and pervasiveness set: OpenBookQA [32], Gender [33], RTE [34], IOI (Indirect Object Identification [35]), Docstring [36], SST2 [37], Winogrande [38],

Reverse [39], Greater Than [40], FEVER [41], zsRE [42], Induction [9], Bool [43], Arithmetic [44], and SA (syntactic agreement [45]). We show the examples of each task in the Table 4.

Table 4: An overview of the datasets of specific tasks.

Task	Example	Label
Winogrande	John moved the couch from the garage to the backyard to create space. The _ is small.	garage
SST-2	hide new secretions from the parental units	negative
RTE	No Weapons of Mass Destruction Found in Iraq Yet. Weapons of Mass Destruction Found in Iraq.	not entailment
Bool	(True AND True) OR False	True
Induction	Vernon Dursley and Petunia Durs	ley
IOI	When John and Mary went to the store, Mary gave a bottle of milk to	John
Gender	So Evan is a really great friend, isn't	he
Docstring	def f(self, files, obj, state, size, shape, option): :param state: performance analysis :param size: pattern design :param	shape
Great Than	The war lasted from 1517 to 15	18
SA	Many girls insulted	themselves
arithmetic	What is (2 - 8) - 4? Answer:	-10
Reverse	[0, 3, 2, 1]	[1, 2, 3, 0]
openbookqa	'question_stem': 'The sun is responsible for', 'choices': {'text': ['puppies learning new tricks', 'children growing up and getting old', 'flowers wilting in a vase', 'plants sprouting, blooming and wilting'], 'label': ['A', 'B', 'C', 'D']},	D
FEVER	Kolhan is a village in the Palghar district of Maharashtra ,	India
zsRE	Angela Merkel's second husband is professor Joachim	Sauer

C Ablation Studies and Additional Results on Free Evolution

C.1 Comprehensive Results of Free Evolution

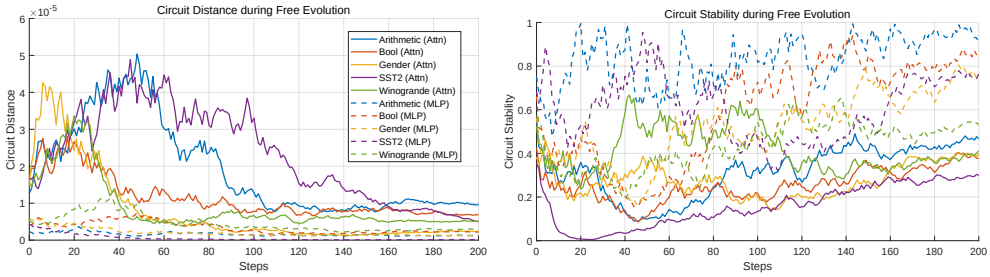
Figure 5 presents the comprehensive evolutionary metrics for five target tasks (Arithmetic, Bool, Gender, SST2, and Winogrande) paired with OpenBookQA as the pervasiveness task, evaluated on both the Mistral-7B and LLaMA3-8B models. These tracked metrics encompass Circuit Distance (CD), Circuit Stability (CS), Task Accuracy (ACC), and Loss. Furthermore, for CD and CS, we separately delineate the evolutionary dynamics of components within the attention and MLP modules, explicitly denoted as $Attn$ and MLP, respectively.

C.2 Impact of Task Type on Free Evolution

As observed in Figure 5, components within the attention and MLP modules exhibit markedly distinct evolutionary patterns. Consequently, we hypothesize that for any given task, the extent of circuit variation during free evolution is heavily influenced by the relative functional contributions of its attention and MLP components. To validate this hypothesis, we designed the following ablation study.

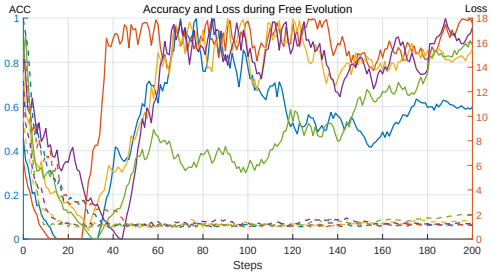
Prior research has established that tasks dominated by MLP components are predominantly associated with stored factual "knowledge" [3](e.g., factual queries regarding nations, geography, prominent figures, and sports), whereas tasks dominated by attention components are generally tied to latent functional "skills" (e.g., induction heads [52] and syntax heads [53]). Building upon this consensus, we select the *Induction* and *Reverse* tasks to represent attention-dominated (skill-centric) tasks, and the *FEVER* and *zsRE* factual datasets to represent MLP-dominated (knowledge-centric) tasks.

Table 5 details the performance across various metrics for these disparate task types on the Mistral-7B model, maintaining OpenBookQA as the pervasiveness task. The Attention and MLP columns represent the percentage of attention/MLP components in the circuit relative to the total number of components in the corresponding computational graph. The results substantiate our initial hypothesis: skill-centric tasks induce substantial migration among attention components, but cause negligible changes in MLP circuit distance. Conversely, knowledge-centric tasks trigger drastic internal parameter updates within MLPs (reflected in stability changes) but result in minimal positional migration of the circuit.

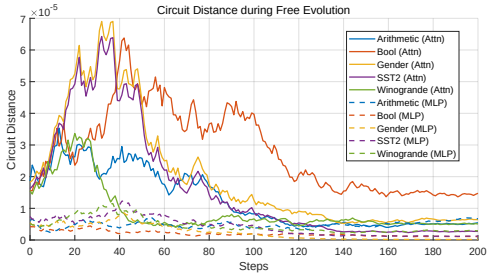


(a) Circuit Distance on Mistral

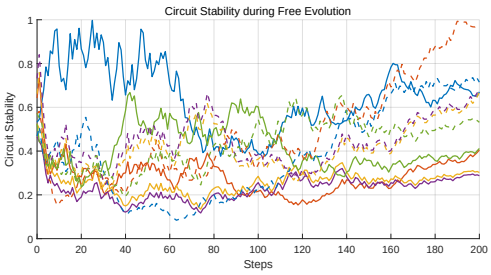
(b) Circuit Stability on Mistral



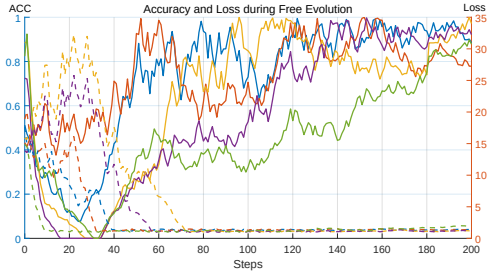
(c) Performance on Mistral



(d) Circuit Distance on LlaMA



(e) Circuit Stability on LlaMA



(f) Performance on LlaMA

Figure 5: line plots of different target tasks on the Mistral-7B model and LlaMA3-8B model in terms of Circuit Distance (CD), Circuit Stability (CS), Task Accuracy (ACC) and loss ($Loss$).

Table 5: Performance of skill-centric tasks and knowledge-centric tasks

Ablation	dataset	Attention(%)	MLP(%)	Δ ACC	Loss	CD of Attn ($\times 10^{-5}$)	CD of MLP ($\times 10^{-5}$)	CS of Attn	CS of MLP
Skill	Induction	75.41	6.25	23.55	1.4217	1.6471	0.8445	0.6577	0.8471
	Reverse	64.33	9.375	33.6	1.2471	1.5421	0.7419	0.5944	0.7619
Knowledge	FEVER	33.71	40.625	15.47	1.1945	1.1474	0.4571	0.8847	0.9492
	zsRE	36.94	56.25	18.47	1.2108	1.2019	0.4138	0.8570	0.9355

C.3 Impact of Pervasiveness Degree on Free Evolution

In the context of LLMs, the optimization of any target task is inherently a multi-objective process accompanied by pervasiveness tasks. Therefore, we conducted an ablation study to investigate the impact of the degree of pervasiveness. Specifically, we selected various concrete tasks, each governing a distinct mechanism, and simulated increasing pervasiveness through their linear superposition (e.g., an ensemble of 10 disparate tasks functioning jointly as the pervasiveness task intrinsically embodies a more "universal" constraint than a single isolated task). Accordingly, we chose Gender, RTE, IOI, Docstring, SST2, Winogrande, Reverse, Greater Than, FEVER, and zsRE as candidate components for the pervasiveness task. Designating Induction as the target task, we observed the circuit evolution on the Mistral-7B model when the total number of concurrent pervasiveness tasks was scaled to 0, 1, 2, 5, and 10.

Table 6 demonstrates that as the degree of pervasiveness increases, the target task exhibits more extensive component migration, while the internally learned information remains largely unaffected. Intuitively, elevated pervasiveness implies a more comprehensive engagement of components across the network during parameter updates, equating to a more constrained and challenging optimization objective. Consequently, the original critical components of the target task typically require more substantial migration to navigate toward the new optimal solution space.

Table 6: Performance of different pervasiveness

task number	Δ ACC	Loss	CD of Attn ($\times 10^{-5}$)	CD of MLP ($\times 10^{-5}$)	CS of Attn	CS of MLP
0	38.99	0.9544	1.4271	0.6355	0.6452	0.6849
1	33.51	1.1146	1.4533	0.6411	0.5391	0.7145
2	31.24	1.1139	1.4951	0.6875	0.5216	0.7351
5	25.17	1.1547	1.6742	0.7422	0.5479	0.6557
10	23.11	1.3049	1.8549	0.8617	0.5311	0.6647

C.4 Impact of Dataset Size on Free Evolution

Similarly, we conducted an ablation study on the impact of dataset size on circuit evolution. We hypothesized that circuit migration might simply be an artifact of insufficient sample sizes, which could lead to uncertain or unstable sampled distributions. To isolate the effect of dataset size while simultaneously eliminating the confounding factor of sample imbalance, we designated the IOI task as the target task. The IOI dataset is a synthetic (procedurally generated) dataset, alleviating concerns that scaling up data samples might introduce real-world biases. Using the Mistral-7B model with OpenBookQA as the pervasiveness task, we evaluated circuit metrics across varying SFT dataset sizes: 500; $2,000$; $5,000$; $10,000$; and $100,000$.

Table 7 indicates that variations in dataset size do not induce significant changes in Circuit Distance (CD); that is, component migration during free evolution is largely independent of dataset size. However, dataset size profoundly impacts the updating of internal information within components. A richer set of training samples facilitates more effective internal information updating, thereby substantially enhancing the robustness of the resulting circuit.

Table 7: Performance of different dataset size

sample number	Δ ACC	Loss	CD of Attn ($\times 10^{-5}$)	CD of MLP ($\times 10^{-5}$)	CS of Attn	CS of MLP
500	28.45	1.2541	1.5749	0.6645	0.5517	0.7541
2000	31.41	1.2037	1.5422	0.6217	0.5916	0.7594
5000	32.04	1.2316	1.5594	0.6559	0.6471	0.8051
10000	32.57	1.1842	1.5207	0.6347	0.6689	0.8067
100000	36.14	1.1554	1.4971	0.6467	0.6953	0.8137

C.5 Impact of Conflict Proportion on Free Evolution

Prior research has established that the quantity of conflicting components among multiple optimization tasks is a critical determinant of multi-task optimization efficacy [18]. Accordingly, we constructed various skill combinations to serve as the pervasiveness task, aiming to observe how an escalating proportion of conflicting components impacts circuit evolution. In our experimental setup, we evaluated the Mistral-7B model using Gender as the target task. We formulated five distinct pervasiveness task combinations that yielded varying proportions of conflicting components relative to the target task, calculated following the methodology in [18]. These combinations and their corresponding conflict proportions are: 5.86% (SST2 + SA), 8.57% (SST2 + SA + RTE), 10.17% (SST2 + SA + RTE + IOI), 12.49% (SST2 + SA + RTE + IOI + Winogrande), and 15.63% (SST2 + SA + RTE + IOI + FEVER).

Table 8 reveals that an increased proportion of conflicting components indeed precipitates significant component migration. As corroborated by previous studies, conflicting components are predominantly polysemantic, encapsulating essential information for multiple tasks simultaneously. When confronted with multi-task optimization, these components struggle to resolve optimal gradient descent directions. Consequently, the language model is compelled to disentangle these multiplexed

semantics, forcing them to migrate into distinct, separate components to satisfy the diverse optimization objectives. Inevitably, this extensive migration necessitates a reorganization of internal neuronal semantics, which subsequently impedes the effective updating of internal information.

Table 8: Performance of different conflict percentage

percentage(%)	Δ ACC	Loss	CD of Attn ($\times 10^{-5}$)	CD of MLP ($\times 10^{-5}$)	CS of Attn	CS of MLP
5.86	29.63	1.2346	1.4377	0.6147	0.5846	0.7841
8.57	28.46	1.3544	1.4872	0.6355	0.5739	0.7694
10.17	26.17	1.3927	1.5239	0.6741	0.5517	0.7614
12.49	24.39	1.4143	1.5567	0.6964	0.5228	0.7259
15.63	22.15	1.5467	1.6741	0.6857	0.4677	0.7118

C.6 Impact of Initial Mastery Level on Free Evolution

Finally, we investigate the impact of the model's intrinsic mastery level of the target skill. This investigation is driven by a compelling hypothesis: if a model has already attained 100% mastery of a task's mechanism (i.e., achieving 100% accuracy), will its corresponding circuit still undergo evolution during subsequent parameter updates?

To explore this, we selected a set of high-mastery tasks (Gender, SST2, Winogrande) and low-mastery tasks (Bool, Arithmetic, SA). High-mastery tasks are defined as those exhibiting a pre-SFT accuracy exceeding 70%, whereas low-mastery tasks fall below 40%. Furthermore, we curated a specialized subset comprising 500 correctly answered samples extracted from the high-mastery datasets, ensuring a rigorous initial accuracy of 100%. Designating these datasets as target tasks and OpenBookQA as the pervasiveness task, we conducted ablation experiments on the Mistral-7B model.

Table 9 demonstrates that tasks with higher initial mastery exhibit a diminished degree of component migration, suggesting that the model has already partially solidified the underlying mechanism. Crucially, however, even for skills mastered at 100%, the circuit still undergoes definitive migration and evolution during parameter updates. This finding compellingly reinforces the conclusion that a circuit is an inherently dynamic property; therefore, a static circuit derived from current parameters cannot reliably dictate or guide future parameter updates.

Table 9: Performance of different mastery

ablation	dataset	Δ ACC	Loss	CD of Attn ($\times 10^{-5}$)	CD of MLP ($\times 10^{-5}$)	CS of Attn	CS of MLP
Mastered	Gender	3.37	1.2417	0.8947	0.9517	0.6759	0.8472
	SST2	22.51	1.1694	1.1167	0.6646	0.8473	0.9618
	Winogrande	14.79	1.2547	0.9382	0.5938	0.7882	0.8611
100% Mastered	Gender	-2.71	1.1544	0.8017	0.8851	0.8543	0.8867
	SST2	-6.88	1.1392	1.0544	0.7419	0.8144	0.9546
	Winogrande	-5.93	1.1582	1.0649	0.8244	0.7967	0.9386
Unmastered	Bool	39.45	1.3347	1.3547	1.1124	0.4985	0.6719
	Arithmetic	32.57	1.3629	1.6691	0.6749	0.4467	0.6637
	SA	29.67	1.2984	1.8932	0.9839	0.5278	0.6565

D Reconciliation and Comparison with Existing Studies on Fine-Tuning Circuit Dynamics

A growing body of literature investigates how internal mechanistic circuits evolve during SFT. However, because existing works evaluate distinct model families, task domains, and fine-tuning setups, they often arrive at seemingly conflicting conclusions regarding whether circuits remain static or dynamically migrate. In this section, we review four representative works and demonstrate how their empirical findings can be unified and reconciled within our comprehensive analytical framework.

Overview of Representative Literature.

- Research [49] employ EAP-IG for circuit discovery on arithmetic tasks, observing that task circuits undergo substantial structural transformations during SFT, with edge-level changes being 2–3$\times$ more pronounced than node-level changes. Based on this drift, they design an adaptive rank-allocation strategy for LoRA.

- Research [48] utilize greedy path patching on an entity tracking task, concluding that the primary circuit components remain structurally fixed before and after fine-tuning, with SFT merely enhancing their causal effect magnitudes rather than altering component positions.
- Research [47] apply linear probing on synthetic toy models (*e.g.*, Tracr and PCFG-trained networks), finding that fine-tuning rarely alters underlying core mechanisms, instead learning minimal "wrapper" modules on top of pre-existing circuits.
- Research [46] conduct circuit discovery on Indirect Object Identification (IOI) under standard SFT, data poisoning, and repair retraining, finding that fine-tuning significantly alters circuit effect strength and rewires internal functional dependencies.

Reconciling Empirical Observations. The empirical findings of research [49, 46] directly validate our core conclusions: fine-tuning induces profound mechanistic shifts, including topological rewiring and continuous changes in component-level causal contributions. Our proposed metric, **Circuit Distance (CD)**, explicitly captures these dynamics by evaluating real-valued contribution shifts across all components, thereby reflecting functional re-assignment, topological rewiring, and causal intensity shifts beyond binary node inclusion/exclusion.

Conversely, the observations in research [48, 47] represent special, localized sub-cases within our framework rather than contradictory evidence:

1. **High Initial Mastery Mitigates Migration (research [48]):** evaluate entity tracking—a task on which pre-trained LLMs already exhibit high native proficiency. As demonstrated in our analysis (Table 9), tasks with high initial mastery require minimal representation re-organization, naturally exhibiting significantly smaller Circuit Distance shifts (CD). Furthermore, their reliance on greedy path patching risks extraction incompleteness, while discrete component selection fails to capture continuous causal drift [28].
2. **Limitations of Synthetic "Wrappers" (research [47]):** The "wrapper" hypothesis itself acknowledges dynamic adaptation, as new components are recruited to form task-specific responses. More importantly, research [47] draw conclusions from synthetic toy models (*e.g.*, compiled RASP networks) evaluated via non-causal probing. Synthetic, single-task models lack the dense, multi-domain parameter distributions of large language models, making it unreliable to extrapolate "wrapper" behavior to full-scale LLM fine-tuning dynamics.

Methodological and Empirical Advantages of Our Work. Compared to existing literature, our work offers two key advantages:

1. **Methodological Rigor and Quantitative Analysis:** By integrating state-of-the-art Logical Circuit Discovery with three novel metrics (**CD**, **CS**, and **CC**), we transition mechanistic analysis from qualitative, single-task observation to fine-grained, quantitative measurement of structural evolution.
2. **Comprehensive Evaluation Spectrum:** We systematically evaluate the broadest spectrum of SFT conditions—including single- vs. multi-task regimes, localized vs. unconstrained updates, high vs. low initial mastery, and skill- vs. knowledge-intensive tasks. This comprehensive coverage allows prior observations to emerge as sub-cases within our unified empirical framework.

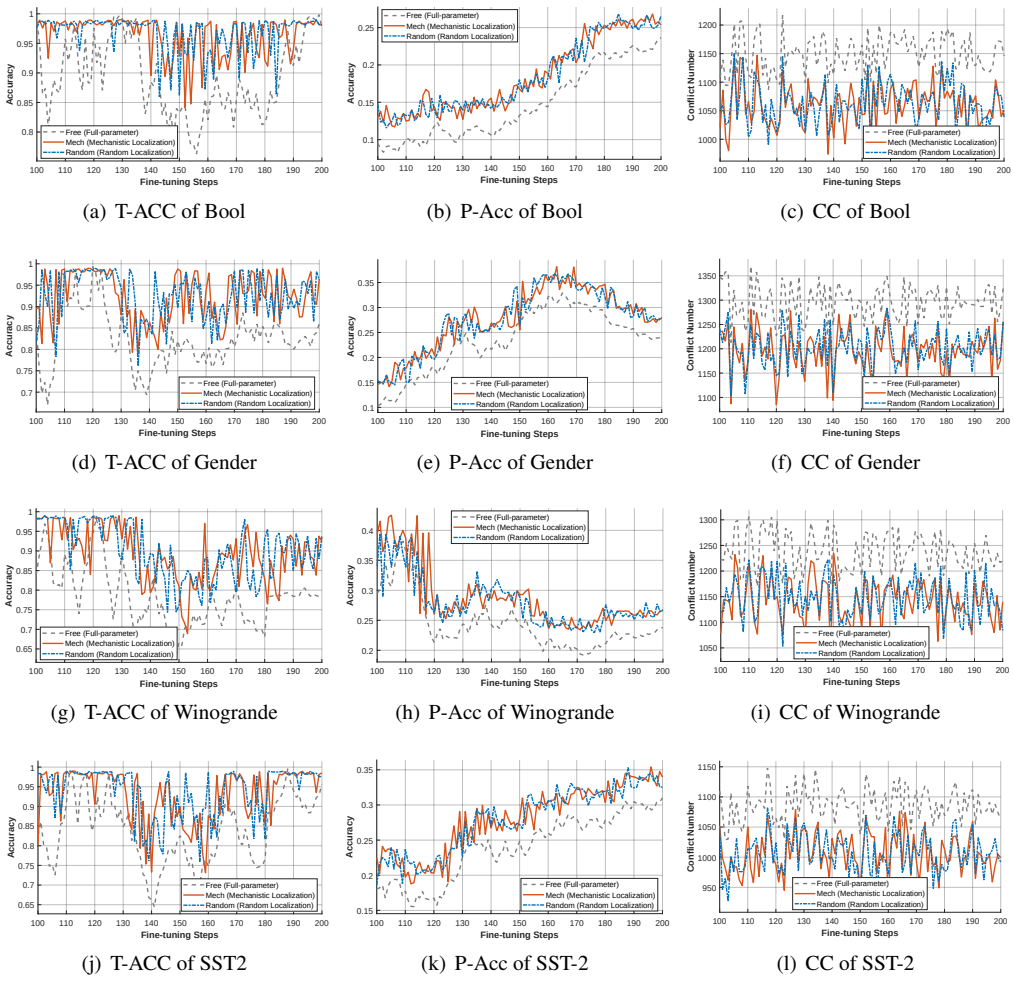
E More Experiments Results with Localization

E.1 More Target Tasks on Three Localization Strategies

Figure 6 details the evolutionary trajectories of the remaining four target tasks across the Accuracy and Circuit Conflict metrics. Corroborating the findings from the Arithmetic task, the localization methods consistently exhibit superior performance relative to the *Free* evolution baseline.

E.2 Impact of Localization on Circuit Distance and Stability

Figure 7 illustrates the performance of the five target tasks concerning Circuit Distance (CD) and Circuit Stability (CS). A pronounced observation is that localization methods precipitate substantially



(a) T-ACC of Bool (b) P-Acc of Bool (c) CC of Bool

(d) T-ACC of Gender (e) P-Acc of Gender (f) CC of Gender

(g) T-ACC of Winogrande (h) P-Acc of Winogrande (i) CC of Winogrande

(j) T-ACC of SST2 (k) P-Acc of SST-2 (l) CC of SST-2

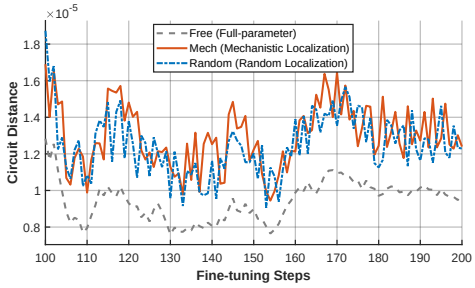
Figure 6: Target Task Accuracy (T-Acc), Pervasiveness Task Accuracy (P-Acc), and Circuit Conflict (CC) of Bool, Gender, Winogrande, and SST-2 Task with localization.

greater component migration compared to free evolution. This strongly implies that the components pinpointed by localization do not constitute the genuinely optimal subset required for adaptation; paradoxically, this artificial restriction renders the evolutionary trajectory even more "incorrect" than if no localization were applied. Concurrently, the sharp decline in CS demonstrates that it becomes markedly more arduous to effectively update internal information within the remaining unfrozen components. Collectively, these phenomena substantiate that current Mechanistic Localization fails to isolate the truly critical components and inadvertently freezes components essential for natural adaptation. It is precisely this misallocation that induces the chaotic volatility observed in the individual circuit metrics.

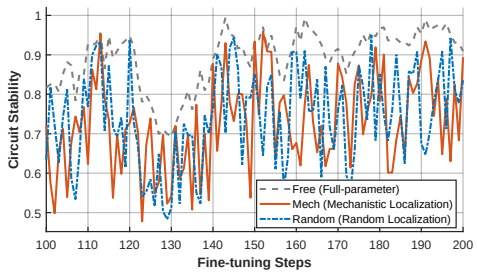
E.3 Impact of Circuit Scale on Localization Efficacy

To further investigate the limitations of static localization, we conducted a systematic interpolation study on the circuit scale. We evaluated task performance (T-Acc and P-Acc), Circuit Distance (CD), Circuit Stability (CS), and Circuit Conflict (CC) across varying circuit capacities: 500, 800, $1,000$, $1,500$, $2,000$, $2,500$, $3,000$, $3,500$, and $4,000$ (corresponding to the total number of unfrozen critical components).

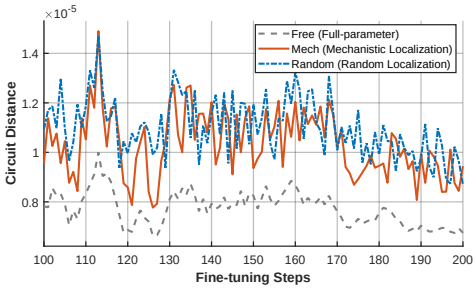
As demonstrated in Table 10, optimal performance across all metrics is achieved when the circuit scale expands to approximately $3,000$–$3,500$ components. At this capacity, the localized subset is sufficiently expansive to encompass those latent components that—while deemed "marginally



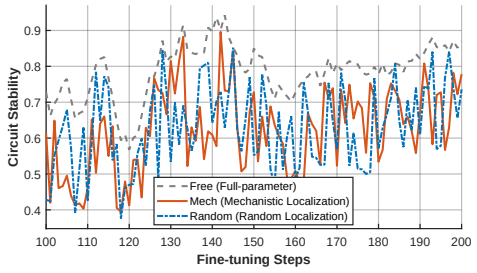
(a) CD of Arithmetic



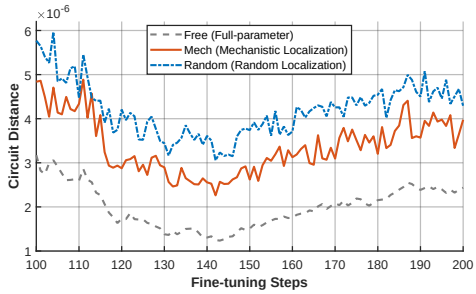
(b) CS of Arithmetic



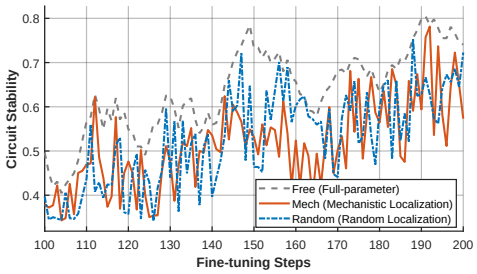
(c) CD of Bool



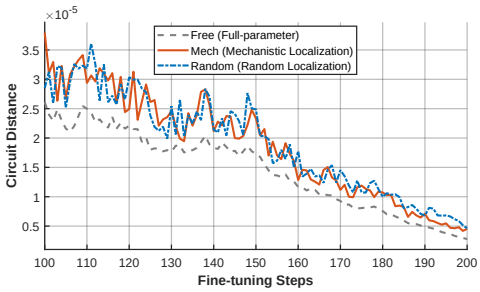
(d) CS of Bool



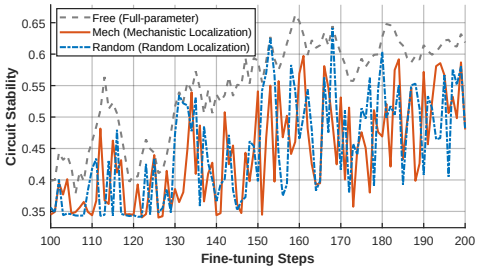
(e) CD of Gender



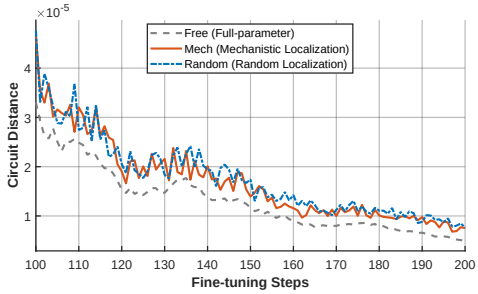
(f) CS of Gender



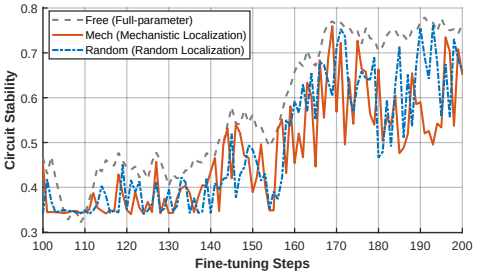
(g) CD of Winogrande



(h) CS of Winogrande



(i) CD of SST2



(j) CS of SST-2

Figure 7: Circuit Distance (CD) and Circuit Stability (CS) of Arithmetic, Bool, Gender, Winogrande, and SST-2 Task with localization.

important" under the current parameter state—prove highly critical for future parameter updates. This highlights a fundamental trade-off inherent in determining circuit scale: an overly restrictive scale reliably isolates components critical to the *current* state but inadvertently discards latent components vital for *future* evolution; conversely, an excessively permissive scale severely compromises the very "interpretability" the mechanism seeks to provide.

Table 10: Performance of different circuit scale in Arithmetic task

Localization	Metric	500	800	1,000	1,500	2,000	2,500	3,000	3,500	4,000
Mech	T-Acc	0.75	0.76	0.73	0.71	0.73	0.73	**0.75**	**0.75**	0.72
	P-Acc	0.26	0.25	0.24	0.25	0.26	0.24	**0.26**	**0.27**	0.25
	$CD(1*10^{-5})$	1.38	1.24	1.28	**1.26**	1.27	**1.33**	**1.35**	**1.32**	1.29
	CS	0.84	**0.86**	0.81	**0.75**	0.88	0.75	**0.70**	**0.65**	**0.61**
	CC	1105	1247	**1067**	1134	1212	**1069**	**1109**	**1142**	**1129**
Random	T-Acc	0.75	0.75	0.74	0.72	0.73	0.72	0.73	0.74	0.73
	P-Acc	0.25	0.25	0.25	0.24	0.27	0.25	0.24	0.25	0.25
	$CD(1*10^{-5})$	1.33	1.21	1.26	1.28	1.27	1.38	1.41	1.36	1.25
	CS	0.85	0.82	0.86	0.74	0.84	0.77	0.60	0.58	0.53
	CC	1006	1212	1075	1123	1201	1077	1215	1194	1168

Crucially, however, at this optimal scale of $3,000$–$3,500$ components, the localized circuit encompasses nearly 80% of the language model's total target components. Consequently, any assertion regarding the inherent "effectiveness of localization" becomes exceedingly tenuous at this juncture, as the tuning process fundamentally regresses toward standard full-parameter SFT.

Additionally, we evaluated the comparative efficacy of the three distinct circuit extraction methodologies outlined in Appendix A. Table 11 delineates the performance disparities among ACDC, EAP, and EdgePruning. It is evident that the EAP approach markedly outperforms the other two alternatives. Synthesizing this empirical finding with our analysis of **RQ2-c** in Section 5.2, a compelling explanation emerges: EAP's reliance on gradient-based estimation aligns inherently with the trajectory of gradient descent. Consequently, this characteristic renders the EAP-derived circuits substantially more compatible with, and adaptive to, future dynamic parameter updates.

Table 11: Performance of different circuit methodologies

Dataset	Circuit	T-Acc	P-Acc	$CD(1*10^{-5})$	CS	CC
Arithmetic	EAP	0.73	0.26	1.27	0.88	1212
	ACDC	0.71	0.22	1.42	0.87	1154
	EdgePruning	0.74	0.25	1.55	0.87	1205
Gender	EAP	0.91	0.28	3.27	0.68	1258
	ACDC	0.88	0.25	4.66	0.75	1239
	EdgePruning	0.84	0.30	3.69	0.61	1349

F Validation in Single-Objective SFT

In Section 4.2.2, we established a critical conclusion: in tasks governed by MLP-dominated circuits, the circuits are inherently more resistant to migration. Consequently, Mechanistic Localization coincidentally retains its guiding significance for future parameter updates, yielding performance metrics significantly superior to both Random Localization and free evolution. Conversely, for tasks governed by Attention-dominated circuits, intense component migration precipitates massive structural discrepancies between the circuit under the current parameter state and circuits under future states. Thus, Mechanistic Localization fails to provide predictive guidance for future parameters, resulting in performance virtually indistinguishable from Random Localization.

To rigorously validate this conclusion, we eliminated the pervasiveness task during the SFT of the Mistral-7B model. This allowed us to observe whether the isolated target task exhibits similar evolutionary patterns under pure single-objective optimization.

Table 12 and 13 unequivocally demonstrates that when WMDP-Bio and Induction are optimized independently as isolated target tasks, the following conclusions persistently hold true:

Table 12: Performance of WMDP-Bio dataset without pervasiveness task

Metric	Localization	Epoch 0	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5	Epoch 6	Epoch 7	Epoch 8	Epoch 9	Epoch 10
ACC	Free	0.36	0.01	0.00	0.00	0.00	0.15	0.26	0.45	0.49	0.59	0.72
	Random	0.36	0.00	0.00	0.04	0.26	0.19	0.35	0.55	0.68	0.72	0.75
	Mech	0.36	0.00	0.00	0.00	0.15	0.38	0.67	0.59	0.77	0.82	0.85
CD	Free	0	0.95	1.12	1.23	1.15	1.08	1.02	0.95	0.98	0.94	0.93
	Random	0	1.34	1.44	1.52	1.59	1.51	1.55	1.35	1.39	1.25	1.21
	Mech	0	1.59	1.66	1.54	1.42	1.39	1.21	1.14	1.19	1.11	1.11
CS	Free	0.78	0.35	0.29	0.21	0.39	0.44	0.64	0.58	0.66	0.75	0.84
	Random	0.78	0.29	0.15	0.34	0.46	0.42	0.51	0.67	0.79	0.75	0.89
	Mech	0.78	0.33	0.29	0.27	0.62	0.35	0.55	0.61	0.79	0.85	0.94
CC	Free	982	1057	1132	1164	1192	1167	1158	1138	1146	1142	1084
	Random	982	1145	1231	1214	1195	1194	1163	1147	1195	1175	1181
	Mech	982	1129	1178	1235	1207	1151	1129	1166	1157	1121	1096

Table 13: Performance of Induction dataset without pervasiveness task

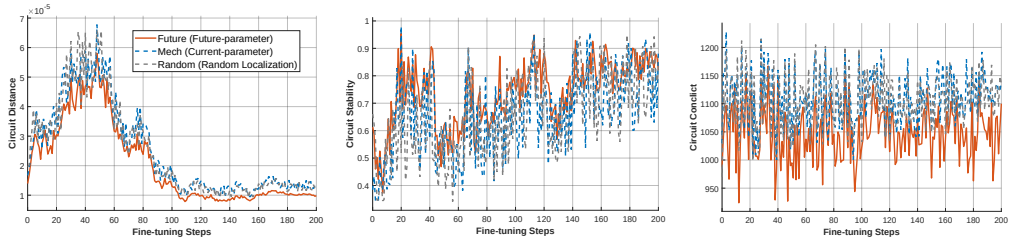
Metric	Localization	Epoch 0	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5	Epoch 6	Epoch 7	Epoch 8	Epoch 9	Epoch 10
ACC	Free	0.45	0.05	0.00	0.00	0.00	0.29	0.64	0.38	0.79	0.66	0.88
	Random	0.45	0.00	0.00	0.00	0.29	0.67	0.49	0.33	0.81	0.67	0.92
	Mech	0.45	0.00	0.00	0.00	0.00	0.54	0.39	0.67	0.81	0.74	0.89
CD	Free	0	1.25	1.33	1.65	1.41	1.58	1.51	1.46	1.48	1.34	1.35
	Random	0	1.67	1.88	1.94	1.92	1.87	1.87	1.83	1.75	1.79	1.69
	Mech	0	1.69	1.92	1.85	1.98	1.75	1.92	1.96	1.88	1.85	1.72
CS	Free	0.63	0.23	0.16	0.31	0.25	0.41	0.48	0.72	0.79	0.85	0.89
	Random	0.63	0.29	0.38	0.19	0.44	0.51	0.67	0.82	0.85	0.89	0.95
	Mech	0.63	0.39	0.34	0.35	0.61	0.59	0.76	0.82	0.89	0.93	0.96
CC	Free	1057	1244	1341	1319	1267	1294	1288	1276	1255	1219	1284
	Random	1057	1209	1185	1167	1139	1194	1205	1185	1157	1167	1139
	Mech	1057	1257	1288	1234	1209	1184	1193	1172	1164	1153	1149

1. **MLP-dominated circuits** are significantly less prone to migration. Therefore, Mechanistic Localization for these knowledge-centric tasks genuinely imparts meaningful guidance for future parameter updates.
2. **Attention-dominated circuits** are highly susceptible to migration, leading to profound structural discrepancies across different parameter states. Consequently, Mechanistic Localization for these skill-centric tasks suffers from severe temporal latency, rendering it ineffective for guiding dynamic updates.

G Extended Analysis of Future Mechanistic Localization and Methodological Comparisons

First, we present the comprehensive circuit metrics—Circuit Distance (CD), Circuit Stability (CS), and Circuit Conflict (CC)—for the Arithmetic task under the *Future Mechanistic* localization paradigm, as illustrated in Figure 8. These detailed metrics further substantiate the profound superiority and stability of the *Future Mechanistic* approach over static baseline methods.

Subsequently, we expand our comparative analysis beyond standard circuit discovery to encompass other predominant Mechanistic Localization methodologies. Specifically, we evaluate gradient-based methods (e.g., WAGLE and DEPN), intervention-based methods (e.g., MEMIT and CLUE), and strictly circuit-centric methods (e.g., CLUE-EAP and CLUE-EdgePruning).



(a) Circuit Distance (b) Circuit Stability (c) Circuit Conflict

Figure 8: Line plots of Future Mechanistic Localization

As demonstrated in Table 14, gradient-associated methodologies consistently yield the optimal CD and CS metrics among all evaluated techniques. Notably, this superior performance persists even for methods that do not rely on standard gradient descent fine-tuning for parameter updates (e.g., MEMIT, which employs closed-form vector editing). The introduction of these strategies of Localization are as follows:

DEPN [1] (Detect and Edit Privacy Neurons) is a framework designed to safeguard against privacy leakage in pretrained language models by localizing and editing specific neurons. The method's core localization component is a novel privacy neuron detector that uses a gradient-based attribution technique. This detector computes a privacy attribution score for each neuron to quantify its contribution to the model's leakage of private information. This is achieved by calculating the cumulative gradient of the output probability with respect to the neuron's activation value, as the activation is gradually changed from zero to its original value.

WAGLE [19] (Weight Attribution-guided LLM Unlearning Framework) is a framework that pinpoints the most influential weights for unlearning through a strategic weight attribution method. The method frames the weight attribution problem as a bi-level optimization (BLO) problem, which allows it to balance unlearning efficacy with utility preservation. The core of the localization process is the derivation of a closed-form attribution score for each weight, calculated using the implicit gradient from the BLO problem. This score's value is determined by combining the gradients from both the forget loss and the retain loss.

MEMIT [54] ((Mass-Editing Memory in a Transformer)) addresses the deletion of factual information by causal tracing, a denoising-based intervention method. This approach relies on the assumption that knowledge is stored in specific, localized components of the network, and can be identified via causal mediation.

KN [4] (Knowledge Neurons) introduces the concept of knowledge neurons to investigate how factual knowledge is stored in pretrained Transformers. As an intervention-based method, it views feed-forward network (FFN) modules as key-value memories. The method utilizes a knowledge attribution technique based on integrated gradients to evaluate the contribution of each neuron to knowledge predictions. By identifying and manipulating (e.g., suppressing or amplifying) these specific neurons, KN demonstrates that interventions on localized neurons can explicitly affect knowledge expression and edit factual knowledge within the model without the need for SFT.

CLUE [18] (Conflict-guided Localization for LLM Unlearning Framework) is a circuit-based localization framework designed to improve the precision of LLM unlearning. It leverages mechanistic interpretability to discover logical circuits corresponding to the forget set and the retain set. CLUE transforms these circuits into Conjunctive Normal Form (CNF) and uses a Boolean satisfiability solver to disentangle the intertwined nodes into three distinct categories: forget nodes, retain nodes, and conflict nodes. By pinpointing the specific function of each node, CLUE enables targeted SFT strategies that significantly enhance forget efficacy while preserving retain utility, avoiding the pitfalls of applying uniform interventions on entangled nodes.

This empirical evidence strongly implies that gradient-based techniques inherently possess greater "foresight" during the parameter update process; they are capable of prospectively identifying a crucial subset of the critical components that will ultimately govern the future parameter state.

Table 14: Performance of different Localization Strategies on Arithmetic Dataset

Categories	Localization	Updating	CD	CS	CC	T-Acc	P-Acc
Gradient	WAGLE	Fine-Tuning	1.31	0.89	1157	0.74	0.25
	DEPN	Fine-Tuning	1.29	0.85	1137	0.72	0.26
Intervention	MEMIT	Editing	1.64	0.74	1344	0.71	0.22
	KN	Editing	1.68	0.72	1315	0.72	0.23
Circuit	CLUE (EdgePruning)	Fine-Tuning	1.55	0.87	1205	0.74	0.25
	CLUE (EAP)	Fine-Tuning	1.27	0.88	1212	0.73	0.26

H Limitations

While this paper derives a series of insightful conclusions by observing transformer circuits throughout the Supervised Fine-Tuning (SFT) process, several limitations remain to be acknowledged:

1. **Inherent Limitations of Circuit Discovery:** The process of circuit discovery itself is notoriously difficult to scale to exceptionally large LLMs and imposes stringent requirements on data quality. Consequently, this computational bottleneck precludes further analysis under massive data and model scaling scenarios, thereby restricting the direct application of our analytical framework in certain real-world, large-scale deployments.
2. **Coupling of Localization and Parameter Update Mechanisms:** Many contemporary Mechanistic Localization methodologies introduce bespoke parameter update techniques paired with their localization strategies; the effects of these two components are rarely strictly independent. Although employing standard SFT as our observational baseline allows us to capture universally applicable and dynamic evolutionary trends, integrating our framework with these specialized, coupled update methods could potentially unveil more granular and nuanced patterns that remain unexplored in this work.

I Broader Impact

This work advances the field of mechanistic interpretability by shifting the paradigm from static structural analysis to dynamic, predictive evolution. By exposing the "illusion of effectiveness" in current methodologies and introducing the necessity of "foresight" in localization, we enable more robust and enduring post-training interventions. This carries profound positive implications for LLM alignment, knowledge editing, and machine unlearning. Specifically, endowing localization with predictive capabilities allows developers to more effectively erase harmful biases, unlearn copyrighted or toxic materials, and rectify factual errors without inadvertently degrading the model's general capabilities through mechanism migration.

However, as with many advanced interpretability and model-editing techniques, our research presents inherent dual-use risks. The ability to precisely locate and persistently update specific task mechanisms could be weaponized by malicious actors. For instance, adversaries could leverage predictive mechanistic localization to inject subtle backdoors, bypass safety guardrails, or manipulate factual knowledge without triggering performance anomalies in standard evaluations. To mitigate these risks, we strongly encourage the community to couple advancements in mechanistic localization with robust, holistic red-teaming and comprehensive safety audits prior to the real-world deployment of edited models.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: We summarize the three conclusions of this paper in Introduction Section.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: Appendix H

Guidelines:

- The answer [N/A] 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: We validate all assumptions in our Experiments Section.

   Guidelines:

   - The answer [N/A] 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: We claim it in Appendix B.

   Guidelines:

   - The answer [N/A] 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: The code url is shown in Abstract.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: We claim it in the Appendix B.

   Guidelines:

   - The answer [N/A] 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: [N/A]

   Justification: The metrics have considered the statistical significance.

   Guidelines:

   - The answer [N/A] 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: We claim it in the Appendix B.

   Guidelines:

   - The answer [N/A] 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: [N/A]

   Justification: There is no research in this paper involved with Ethics.

   Guidelines:

   - The answer [N/A] 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: We discuss the potential future research direction in Appendix I.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We show the URL of pre-trained LLMs for checking.

    Guidelines:

    - The answer [N/A] 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: All assets are properly credited.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: No new assets.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: No human subjects experiments.

Guidelines:

- The answer [N/A] 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: [N/A]

    Justification: No human subjects experiments.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Only for formatting.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.