Knowledge Tracing and Editing in Language Models: Misalignment, Analysis, and a Roadmap

Anonymous ACL submission

Abstract

Large language models encode substantial factual knowledge in their parameters, but this knowledge is difficult to inspect and to update reliably. Two closely related research directions address this challenge: knowledge tracing, which aims to reveal how specific knowledge is represented and used inside the model, and knowledge editing, which aims to modify targeted facts while preserving general capabilities. This survey examines how these areas connect in practice and argues that the link is often unreliable and misaligned. We organize major tracing and editing approaches through an intervention oriented lens, synthesize evidence on where the connection breaks, and outline directions for tighter integration between explanation and intervention in future systems.

1 Introduction

Language models encode vast amounts of factual and relational knowledge implicitly within their parameters, enabling strong performance on knowledge-intensive tasks (Pan et al., 2023; Chen et al., 2022; Hu et al., 2023; He et al., 2023). However, parametric knowledge is difficult to inspect, localize, and reliably update, limiting transparency and controllability (Shi et al., 2025b; Singh et al., 2024). These challenges are particularly critical in high-stakes domains such as healthcare, law, and education, where incorrect or outdated knowledge can undermine trust and safety (Wang et al., 2025d; Chu et al., 2025; Zhang et al., 2025). Therefore, how to construct trustworthy and reliable models is a key issue in the current era of large models.

In response, two closely related research directions have emerged: *knowledge tracing* (KT) and *knowledge editing* (KE). Knowledge tracing (Nostalgebraist, 2020; Abnar and Zuidema, 2020) aims to diagnose the black box of LLMs by uncovering how specific pieces of knowledge are represented, processed and causally utilized within model architectures. Knowledge editing (Meng et al., 2022; Mitchell et al., 2022a) acts as a prescriptive intervention: it seeks to modify targeted factual associations (e.g., correcting outdated statistics, removing misinformation) without overhauling the model's pre-trained parameters or general capabilities. Conceptually, these two fields are inherently complementary: tracing provides the "map" of where knowledge resides, and editing uses that map to execute precise "repairs".

Yet in practice, this synergy remains fragile. A persistent misalignment separates tracing and editing: *editing methods frequently rely on localization assumptions that are only partially supported by tracing, while tracing research rarely incorporates feedback from editing's real interventions.* This disconnect has tangible consequences: conclusions from knowledge tracing are often task-specific, architecture-dependent, and difficult to transfer to editing scenarios. Conversely, knowledge editing faces pervasive side effects, including interference with related facts, degradation of general capabilities, and instability under repeated or large-scale updates (Thede et al., 2025; Gu et al., 2024; Yan et al., 2025; He et al., 2025; Huang et al., 2024; Li et al., 2024a), precisely because tracing insights fail to guide robust intervention design.

Prior surveys (Wang et al., 2025d, 2024b; Mazzia et al., 2025; Yao et al., 2023; Zhang et al., 2024b; Li et al., 2024c; Hase et al., 2024) have summarized advances in tracing or editing in isolation, but none have centered on the gap between explanation (tracing) and intervention (editing) as a unifying lens. In this survey, we treat KT and KE as stages of a *KT→KE pipeline*, as illustrated in Figure 1, which fails at three points. First, there is an *objective mismatch*: tracing highlights sensitivity and functional participation, whereas editing requires stable controllability under intervention. Second, there is a *mechanism mismatch*: editing can re-route or distort the computations that trac-

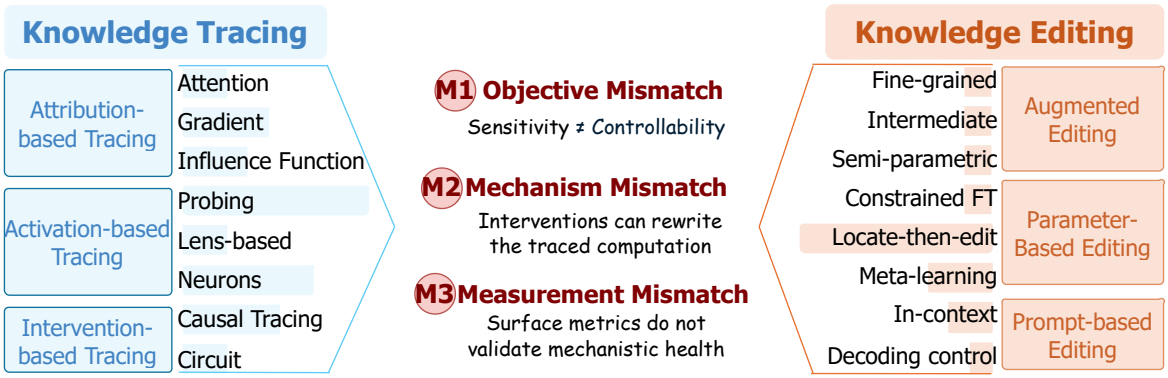


Figure 1: Overview of Mismatches in Knowledge Tracing (KT) and Knowledge Editing (KE) in LLMs. The blue bars indicate how many Editing paper cite each type of Tracing and the orange bars represent the reverse.

ing identifies, even when behavioral edit success appears high. Third, there is a *measurement mismatch*: standard behavioral metrics can miss latent mechanistic degradation and therefore cannot close the loop between tracing and editing.

Concretely, we make three contributions. First, we reorganize tracing methods by what they provide for intervention and categorize editing methods by their *control regimes*. Second, we formalize the above three mismatch points as a minimal typology for the KT–KE interface. Third, we provide complementary micro- and macro-level evidence showing that traced targets can be reproducible yet offer weak control leverage, and that behaviorally successful edits can still induce mechanistic drift. Together, these results motivate a closed-loop research agenda where editing serves as an interventional validator for tracing, and tracing serves as a mechanistic constraint for editing.

Paper Roadmap. Section 2 reviews KT and KE methods, and clarifies the *KT→KE interface*: what tracing provides versus what editing requires. Section 3.1 formalizes three recurring failure points in that interface, and Sections 3.2–3.4 ground them with micro-level mechanistic diagnostics and macro-level ecosystem evidence. Finally, we outline a roadmap toward a closed-loop synergy between tracing and editing.

2 Knowledge Tracing and Editing: Methods and Current Practice

This section reviews recent advances in knowledge tracing (Sec. 2.1) and knowledge editing (Sec. 2.2), and summarizes the current practice from an intervention-oriented perspective. Figure 10 in Appendix F provides an architectural landscape of the two lines of work. Throughout, we emphasize what each method family *outputs* in practice (e.g., candidate components, causal effect maps, update operators), as these outputs form the implicit interface that later sections examine.

2.1 Knowledge Tracing

Knowledge tracing seeks to pinpoint *where* specific information is encoded in the model and *how* such information is processed to produce a prediction. Operationally, tracing methods differ mainly in the type of evidence they produce and the level at which they localize knowledge. We group them into *attribution-based*, *activation-based*, and *intervention-based* methods.

Attribution-based Methods. Attribution methods quantify how input features affect model predictions, producing token- or span-level importance scores in the *input space*. Early work used *attention-based attribution* (Abnar and Zuidema, 2020; Clark et al., 2019; Ayyar et al., 2025), treating attention weights as token-importance scores, but later studies showed that attention is not inherently interpretable and can vary without changing outputs (Jain and Wallace, 2019; Liu et al., 2022; Pruthi et al., 2020). This motivated *gradient-based approaches* (Nielsen et al., 2022; Ancona et al., 2018; Smilkov et al., 2017), which use local derivatives as feature attributions but suffer from saturation, motivating path-integral methods such as Integrated Gradients (Sundararajan et al., 2017; Kapishnikov et al., 2021; Lundstrom and Razaviyayn, 2025). At the data level, *influence-function* (Koh and Liang, 2017; Koh et al., 2019) estimates how upweighting a training point changes a test prediction, but recent work (Li et al., 2025b) finds that

classical influence approximations break down for modern LLMs. In practice, attribution methods are lightweight and widely applicable, typically providing saliency-style evidence that can serve as a first-pass hypothesis for deeper localization.

Activation-based Methods. Activation-based approaches localize knowledge in internal representations, shifting from important *inputs* to informative *hidden states*. *Probing methods* (Conneau et al., 2018; Tenney et al., 2019; Belinkov, 2022) use auxiliary classifiers to decode conceptual attributes from hidden states; linear probes (Alain and Bengio, 2016; Dong et al., 2025) are simple and interpretable, while nonlinear probes (Gairola et al., 2025; Hewitt and Liang, 2019) add capacity but risk overfitting. *Lens-based methods* decode intermediate activations via Logit Lens (Nostalgebraist, 2020; Neo et al., 2025), Tuned Lens (Belrose et al., 2023), Entropy Lens (Ali et al., 2025), Backward Lens (Katz et al., 2024) and extensions to encoder–decoders and vision models (Langedijk et al., 2024; Takatsuki et al., 2025) to trace how predictions evolve across layers. *Knowledge-neuron methods* (Dai et al., 2022; Wang et al., 2024d; Chen et al., 2025b) attempt to link neurons to specific facts but face polysemanticity (Mu and Andreas, 2020), motivating *sparse autoencoders* (Huben et al., 2024; Braun et al., 2024; Bricken et al., 2023) which produce monosemantic features and consistent concept dictionaries (Fel et al., 2025; Shi et al., 2025a; Arad et al., 2025; Kang et al., 2025). In practice, activation-based methods often output *candidate layers/neurons/features* and *representation-level trajectories* that suggest where the model encodes and transforms knowledge, and are used as inputs to downstream interventions.

Intervention-based Methods. Intervention-based tracing treats neural networks as causal systems and tests whether internal states are necessary or sufficient for a behavior via controlled manipulations (Vig et al., 2020; Geiger et al., 2025). *Causal tracing* (activation patching (Meng et al., 2022; Vig et al., 2020; Zhang and Nanda, 2024; Dumas et al., 2025) compares clean and corrupted runs to locate states whose restoration fixes model outputs. Extending this idea, *circuit* methods (Dumas et al., 2025; Conmy et al., 2023; Hsu et al., 2025; Haklay et al., 2025; Lan et al., 2024) isolate minimal head/MLP subgraphs underlying behaviors, and recent work (Yao et al., 2024; Merullo et al., 2024; Lan et al., 2024) shows that models often reuse shared subcircuits. Theoretical work on *causal abstraction* (Geiger et al., 2021, 2025; Hu and Tian, 2022; Geiger et al., 2022, 2024; Wu et al., 2023; Tan, 2023) formalizes these ideas, while recent results (Sutter et al., 2025) highlight that unconstrained abstraction maps can trivialize such claims, motivating stricter structural assumptions. In practice, intervention-based methods output *causal effect maps* and *causal hypotheses about which modules carry which information*, and they provide the closest-to-interventional evidence that is often assumed to be actionable for editing.

2.2 Knowledge Editing

Knowledge editing aims to update a model by inserting, correcting, or removing specific facts without retraining, ideally achieving reliability, locality, generality, and scalability to many edits. Editing approaches differ by where the update is stored and how it is applied during inference. We group them into parameter-based, augmented, and prompt-based editing.

Parameter-based Editing. Parameter-based methods treat pretrained weights as the main storage of knowledge and rewrite them via targeted updates. *Constrained finetuning* (Gekhman et al., 2024; Zhu et al., 2020; Li et al., 2024b) applies regularized updates to specific layers. It suits narrow edit sets by not assuming fact localization, though broad gradient propagation often introduces interference across numerous edits. *Locate-then-edit* approaches (Meng et al., 2022; Wang et al., 2025b; Fang et al., 2025; Meng et al., 2023; Li et al., 2024b; Gupta et al., 2024) instead identify the layers most tied to a fact and modify only those components. This improves locality and interpretability but is brittle when knowledge is distributed or entangled. *Meta-learning* editors (Mitchell et al., 2022a; Tan et al., 2024; De Cao et al., 2021) learn to predict weight updates from an edit specification, enabling fast and scalable edits but adding an auxiliary model whose reliability depends on distributional coverage. In practice, parameter-based editing offers persistent, low-latency changes without extra inference machinery, but it requires white-box access, is prone to interference at scale, and has limited reversibility.

Augmented Editing. Augmented approaches freeze backbone weights and localize updates in

auxiliary mechanisms that can be enabled, disabled, or replaced. *Fine-grained methods* (Wang et al., 2025c; Huang et al., 2023; Bayat et al., 2025; Zhao et al., 2025; QIU et al., 2024; Lee et al., 2025; Kim et al., 2018; Subramani et al., 2022; Pascual et al., 2021; Rimsky et al., 2024; Li et al., 2023; Konen et al., 2024; Stolfo et al., 2025; Todd et al., 2024) steer individual neurons or feature directions, enabling highly targeted edits but requiring sparse, monosemantic feature spaces (e.g., SAEs) to avoid unintended effects. *Intermediate approaches* (Yu et al., 2024; Li et al., 2025a; Wang and Li, 2024) attach trainable, parameter-efficient modules (e.g., adapters) to frozen layers, allowing modular, context-dependent updates at the cost of added inference overhead. *Coarse-grained semi-parametric methods* (Mitchell et al., 2022b; Hartvigsen et al., 2023; Wang et al., 2024a; Zhu et al., 2025; Wang et al., 2025a) store mutable knowledge in external key-value memories that retrieve and override internal representations during inference, improving transparency and reversibility while increasing system complexity. Overall, augmented editing trades architectural simplicity for modularity, scope control, and easier rollback.

Prompt-based Editing. Prompt-based methods keep parameters fixed and apply edits through inputs or decoding control. *In-context methods* (Zheng et al., 2023; Qi et al., 2025; Qiao et al., 2024; Wang et al., 2024c) supply edited facts as examples, instructions, or rationales, allowing the model to adopt updated rules on the fly. *Prompt-level memory* (Wang et al., 2024c; Chen et al., 2025a) stores corrections externally and retrieves them into future prompts. These edits are lightweight but ephemeral: they work only when included in the prompt and consume context length. A complementary direction frames editing as *decoding control* (Wang et al., 2024e; Sun et al., 2024), constraining generation so outputs obey specified knowledge constraints, for example by down-weighting obsolete facts (Sun et al., 2024)by pruning reasoning paths that violate a newly introduced rule. (Wang et al., 2024e). Overall, prompt-based editing is model-agnostic (works for black-box APIs), cheap per edit (no optimization or new parameters), and highly flexible. In practice, prompt-based editing is model-agnostic (works for black-box APIs), but it is not persistent by default and typically offers weaker locality and generalization guarantees.

2.3 From Tracing to Editing: What Tracing Provides and What Editing Requires

Rather than treating knowledge tracing (KT) and knowledge editing (KE) as two independent toolkits, current practice implicitly forms a pipeline: KT proposes *targets* (where/what to intervene on), KE applies an *operator* (how to change the model), and evaluation determines whether the update is *acceptable* (what counts as success). Making it explicit clarifies why a method can be diagnostically informative yet operationally unreliable.

What tracing typically provides. Tracing methods predominantly deliver *sensitivity evidence* about a model's computation on a given prompt. This includes saliency or attribution signals over tokens, layers, neurons, or learned features; ranked candidate components (e.g., attention heads, MLP blocks, feature directions) that appear implicated in producing the output; and, for intervention-based tracing, causal effect maps from controlled activation interventions (e.g., patching) that indicate where restoring information repairs a behavior. These deliverables are valuable for identifying *where the computation is fragile or influential*, but they are not designed to guarantee that the highlighted units are stable *control handles* under subsequent interventions.

What editing requires. Editing methods, in contrast, require *intervention-level guarantees*. First, the chosen target must be *controllable*: intervening on it should reliably move the model toward the intended factual state across prompts and contexts, not merely perturb outputs. Second, edits must satisfy *locality and robustness*: they should minimize unintended changes to related facts and general capabilities, and remain stable under repeated updates or large edit sets. Third, editing demands *validation criteria* that reflect not only surface behavior (e.g., edit success and locality on held-out prompts), but also whether the update preserves intended computation when mechanistic drift is a concern (e.g., avoiding shortcut formation or contextual collapse).

Current practice and implicit interface assumptions. Many locate-then-edit workflows treat tracing outputs as actionable coordinates for editing, implicitly assuming that (i) localized components provide stable leverage across prompts and contexts, (ii) the edit operator modifies knowledge without substantially rerouting computation, and

Seq Setting		Sin Setting	
Neurons	**Freq.**	**Neurons**	**Freq.**
(30, 11268)	24	(30 , 11268)	24
(30 , 11430)	20	(29 , 8499)	24
(30 , 11920)	18	(28 , 2856)	23
(30 , 10321)	15	(30 , 7752)	20
(31 , 7096)	12	(30 , 2458)	19
(31 , 4964)	10	(30 , 14069)	18
(31 , 12111)	10	(31 , 12732)	17
(31 , 5463)	10	(30 , 1441)	17
(31, 154)	9	(30 , 13883)	17
(5 , 7012)	9	(30 , 8061)	17

Table 1: Top recurrent neurons identified, test on 50 data. Seq reflects coupled editing-tracing stability, while Sin indicates the intrinsic importance of neurons across independent data samples.

(iii) standard behavioral metrics are sufficient to validate the update. Section 3.1 examines these assumptions by a pipeline view of *target selection*, *intervention*, and *validation*, and characterizes where and why the tracing→editing connection can fail.

3 Misalignment Between Knowledge Tracing and Editing

The preceding sections established a conceptual complementarity: knowledge tracing diagnoses *where* and *how* a fact participates in model computation, whereas knowledge editing aims to *control* and *update* that knowledge through interventions. Yet in current practice, the KT→KE connection is often treated as a plug-and-play interface: traced components are used as edit coordinates, edits are applied with a chosen operator, and success is judged mostly by behavioral metrics. In this section, we show that the interface breaks in systematic ways, using targeted micro-level diagnostics and macro-level ecosystem evidence.[1]

3.1 The tracing to editing pipeline and three mismatches

We organize the tracing–editing gap by *where* it occurs in a standard KT→KE workflow:

$$\underbrace{\textbf{Target selection}}_{\text{where to intervene}} \rightarrow \underbrace{\textbf{Intervention}}_{\text{how to edit}} \rightarrow \underbrace{\textbf{Validation}}_{\text{behavioral \& mechanistic}}$$

This pipeline view turns "misalignment" into three concrete breakpoints (Figure 1):

[1]The papers we collected (130 Tracing / 90 Editing) are primarily from NeurIPS, ICLR, ICML, ACL, AAAI, and EMNLP (2022–2025). Appendix A provides collection details.

Metric	Sequential	Single
Mean Δgt_prob	-2.37×10^{-7}	-9.19×10^{-8}
Positive Δgt_prob Ratio	0.44	0.42
Argmax Flip Rate	0.22	0.14
Avg #Edited Neurons	11.24	12.26

Table 2: Comparison of sequential (Seq) and Single (Sin) neuron tracing under knowledge editing. Negative Δgt_prob and weak correlations indicate a misalignment between traced neuron importance and effective knowledge editing.

Objective mismatch at target selection: sensitivity $\neq$ controllability. Tracing is diagnostic: it identifies components that *participate* in producing an output (sensitivity). Editing is prescriptive: it requires components that provide stable *leverage* under intervention (controllability). Thus, a component can be causally involved yet still be a poor control handle.

Mechanism mismatch at intervention: edits can rewrite traced computation. Even when tracing isolates a causal pathway for a fact, an edit may introduce a shortcut that reroutes inference (*causal reinforcement*), or gradually erode contextual reliance on unrelated inputs (*causal collapse*). Hence, behavioral success on edited prompts can coexist with mechanistic drift.

Measurement mismatch at validation: surface metrics $\neq$ mechanistic health. Standard editing metrics (e.g., edit success, locality) quantify external behaviors but do not verify whether internal inference pathways remain intact. Conversely, tracing outputs are rarely validated by whether interventions at traced coordinates yield the intended change without side effects. This missing mutual validation keeps the KT→KE loop open.

The following subsections ground each breakpoint with representative micro-level diagnostics and macro-level ecosystem evidence.

3.2 Objective mismatch: sensitivity is not a control handle

A common practice in locate-then-edit pipelines is to treat traced units as actionable edit coordinates. However, tracing outputs are designed to reflect *participation* in the current computation, whereas editing requires *stable leverage* under intervention. We illustrate this mismatch with a neuron-level diagnostic case study.

Neuron-level diagnostic (Knowledge Neurons). We use LLAMA3-CHAT-8B and sample 50 fac-

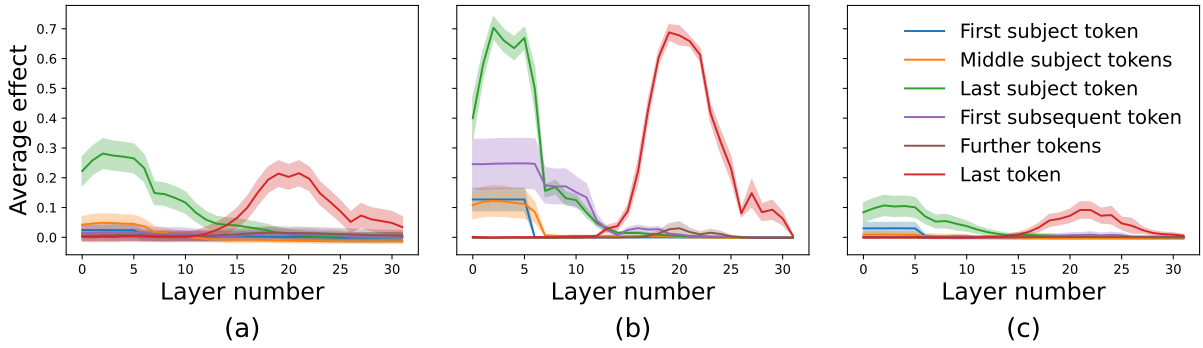


Figure 2: Average effect of MLP blocks. (a) Before Editing, Causal trace on 100 Matched-Data. (b) Post-Edit model with editing on 100 Matched-Data. (c)Post-Edit model with editing on 100 Independent-Data. More Post-Edit model results shows similar trend. Full visualization are provided in Figure 5 – Figure 9 in Appendix E.

tual prompts from KNOWN (Meng et al., 2022). We apply Knowledge Neurons (KN; Dai et al., 2022) to identify top-ranked MLP neurons associated with the ground-truth answer probability via integrated gradients, and intervene on the activations of these neurons during generation. To distinguish intrinsic recurrence from intervention-coupled effects, we compare **Single (Sin)** (trace/intervene independently per example) and **Sequential (Seq)** (accumulate interventions over a stream of examples). We measure mean Δp_{gt} (average change in the model-assigned probability of the ground-truth token), Pos-Δp_{gt} ratio (fraction of cases with $\Delta p_{\text{gt}} > 0$), argmax flip rate (fraction of cases whose top-1 prediction changes after intervention), and the number of edited neurons (intervention scope per instance).

Observation. The traced neurons are highly recurrent and concentrate in late decoding layers (Table 1), suggesting non-trivial localization rather than randomness. Yet intervening on these neurons does not translate into reliable factual gains: mean Δp_{gt} is near-zero and slightly negative, and Pos-Δp_{gt} remains below 50% under both regimes (Table 2). Meanwhile, Seq increases behavioral sensitivity (higher argmax flip rate than Sin), indicating that traced units can affect *what the model says* without providing stable control over *which fact is expressed*.

Implication. Neuron-level tracing can be reproducible and behaviorally influential, but reproducibility and sensitivity alone are insufficient for selecting robust edit coordinates. This is a concrete instance of the objective mismatch at the target-selection breakpoint.

3.3 Mechanism mismatch: edits can reroute the traced computation

Even when editing achieves high behavioral success, the post-edit model may rely on different internal pathways than those suggested by pre-edit tracing. This section uses causal tracing (Meng et al., 2022) to show that edits can introduce shortcut reinforcement on edited data and collapse contextual reliance on unrelated inputs.

Causal diagnostic (AlphaEdit + causal tracing). We fix a trace set $\mathcal{T}$ of 100 samples for causal-tracing analysis and construct edit sets $\mathcal{E}$ with $|\mathcal{E}| \in \{100, 300, 500\}$ for intervention and behavioral evaluation. We use LLAMA3-CHAT-8B and apply AlphaEdit (Fang et al., 2025) (The Sota Editing method) and compute causal effect maps via activation patching following Meng et al. (2022). We consider **Matched-Data (MD)** where $\mathcal{E} = \mathcal{T}$ and **Independent-Data (ID)** where $\mathcal{E} \cap \mathcal{T} = \emptyset$, while causal traces are always computed on $\mathcal{T}$.

Observation. Under Matched-Data, post-edit causal maps show an amplified influence of the **first subsequent token** (Figure 2a vs. Figure 2b), consistent with a strong injected signal that propagates forward and stabilizes the edited output via a shortcut (*causal reinforcement*). Under Independent-Data , causal influence shifts toward over-reliance on the **last subject token** while earlier context contributions diminish (Figure 2c), yielding a systematic narrowing of causal support that we term **causal collapse**; the pattern strengthens as edit scale increases (Figures 7–9 in Appendix E.

Implication. Locate-then-edit interventions can succeed on edited prompts while reshaping inference pathways, producing on-target shortcut re-

Setting	Data Scale	ES	LC
Matched-Data	100	-	0.640
E-100	100	0.192	0.480
E-300	300	0.294	0.160
E-500	500	0.488	0.220

Table 3: Editing success (ES) and locality scores (LC). Locality drastically degrades as the number of edited facts exceeds 100, indicating a loss of model robustness. E-300 means edit on 300 edits in Independent-Data and evaluate the locality on 100 Matched-Data.

liance and off-target contextual collapse.This reflects a mechanism-level mismatch between "what tracing identifies" and "what editing produces".

3.4 Measurement mismatch: surface metrics lag behind mechanistic degradation

A third break point lies in *validation*. In most editing pipelines, success is primarily assessed by behavioral scores such as edit success and locality. At the same time, tracing outputs are rarely evaluated by whether interventions at traced coordinates reliably deliver the intended change with minimal side effects. As a result, the KT→KE loop often remains open: surface metrics may pass while mechanistic drift accumulates, and tracing evidence is not routinely stress-tested for editing. We illustrate this mismatch with a micro-level diagnostic and complementary macro-level ecosystem evidence.

Micro evidence (early warning beyond locality). Table 3 shows that locality can remain moderately high at `E-100` (LC=0.480) even when causal tracing already reveals collapse tendencies (Figure 2c). As edits scale up, this latent mechanistic erosion reaches a tipping point where locality drops sharply (e.g., LC=0.160 at `E-300`). Thus, standard locality metrics can *lag* behind mechanism-level failure, whereas causal traces provide earlier signals of robustness loss.

Macro evidence (open-loop validation in the ecosystem). Publication and influence patterns suggest that validation remains largely open-loop across the KT/KE ecosystem (Figure 3). Tracing established an early foothold and shows steady growth (Figure 3a), whereas editing expands rapidly after 2022 across heterogeneous mechanisms. A quadrant analysis illustrated in Figure 3b using Relative Citation Index and Citation Velocity (detailed in Appendix C) further indicates an imbalance in sustained influence: tracing papers dominate high-impact and high-momentum regions, while editing work is more concentrated in lower-momentum areas. This gap is reinforced by a tooling asymmetry: unified editing toolkits standardize workflows and make behavioral metrics easy to reproduce, whereas tracing tools remain fragmented and less plug-and-play, limiting systematic comparison and transfer across models and edited settings (Table B). Finally, cross-category citation flows (Figure 3c) reveal a strongly asymmetric dependency: editing heavily cites foundational tracing methods, while tracing only sparsely cites editing. Analysis of recent editing paper (Table 5) shows that Parameter-Based edits relies more on tracing, while the other two categories rely less, though tracing could still guide them which is a potential largely unexplored. Together, these patterns reinforce localization as the dominant mental model for knowledge manipulation, even when localization signals are not routinely validated for stability under intervention.

Implication. Both micro and macro evidence point to a validation gap: behavioral metrics do not reliably certify mechanistic health, and tracing outputs are rarely tested for edit-actionability, keeping the KT→KE pipeline open-loop.

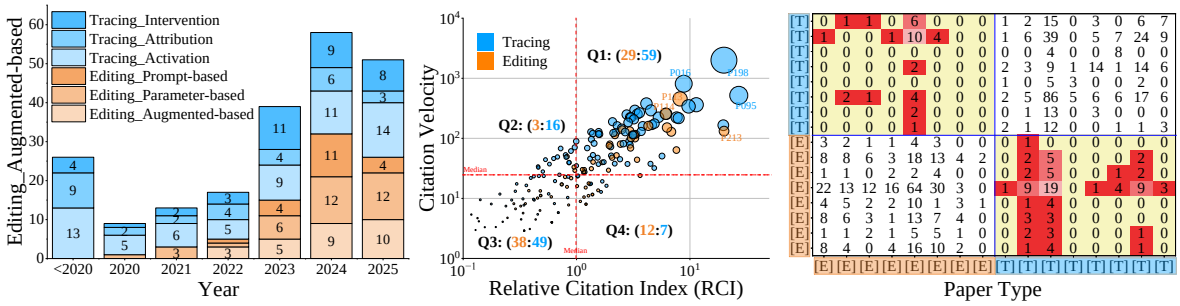
3.5 Takeaway: sensitivity is not manipulability

Across the three pipeline breakpoints, a consistent message emerges: *tracing provides sensitivity signals about participation, but editing requires manipulable levers that preserve mechanistic health under intervention*. Closing the gap needs (i) tracing criteria that explicitly account for stability under intervention, (ii) editing objectives and evaluations that incorporate mechanistic validation rather than relying solely on surface success/locality.

4 Toward a Synergistic Roadmap

We outline four actionable directions to close the loop: (i) make tracing *edit-validated*, (ii) make editing *mechanism-preserving*, (iii) use tracing to *select* editing regimes, and (iv) enable *lifelong* updates with mechanistic monitoring.

(1) Edit-validated Tracing: from sensitivity maps to control handles. Addresses: target selection + validation. Tracing should be judged not only by reproducible localization, but by whether the identified units become *reliable control handles*



(a) Annual Publication Trend (b) Influence vs. Momentum (c) Citation Relation between each types

Figure 3: Macro evidence of mismatch between knowledge tracing and knowledge editing. (a) Paper Count for different type. (b) Categorization of Research Influence: A Quadrant Analysis of Momentum and Status. RCI represents the long-term academic standing of a paper within its field, whereas Citation Velocity indicates its impact rate. (c) Cross-category citation analysis.(Detail for Quadrant and Citation in Appendix D)

under editing. A tracing hypothesis should therefore include an *interventional test suite*: apply a small set of edit operators (e.g., activation steering, targeted weight edits) at the proposed coordinates and check for the predicted behavioral change with strong locality and stability. Editing thus serves as a *falsification probe*: those that cannot support controlled edits are still informative about model behavior, but cannot guide interventions.

(2) Mechanism-preserving Editing: Keep causal integrity, not just surface performance. Addresses: intervention design. Editing objectives should penalize mechanistic drift such as shortcut reinforcement or contextual collapse . Beyond edit success and locality, editors can add *mechanistic constraints* from tracing, for example, matching pre/post causal, effect maps on a fixed probe set, or preserving influence profiles of key contextual features on unrelated prompts. These can be implemented as lightweight regularizers and used as early-warning signals during large-scale edits, ensuring the edit changes the intended association while keeping surrounding computation stable.

(3) Tracing-guided Regime Selection: a navigational signal for hybrid edits. Addresses: target selection. Our results suggest that "where the model is sensitiv" does not uniquely determine "how to edit." Rather than treating parameter-, augmented-, and prompt-based edits as interchangeable, tracing can drive a controller that selects the *editing regime* based on the representation profile of the target knowledge. Knowledge that is causally integrated into multi-step inference (broad, context-dependent support) may warrant parameter-level edits with strong mechanism-preservation constraints, while highly localized facts may be better handled by reversible augmentation (memories/adapters) or prompt-level control. Tracing becomes a *decision signal* about permanence, not a direct prescription of a single layer/neuron.

(4) Lifelong Closed-loop Updating: continual edits with monitoring and rollback. Addresses: intervention design + validation. Deployments require many updates, where cumulative interference and latent mechanistic erosion are first-order risks. A synergistic framework should treat editing as continual: maintain a persistent probe set for mechanistic health, periodically re-trace to detect representational drift, and support rollback via modular updates (e.g., removable memories/adapters when feasible). The system then optimizes the *trajectory* of updates rather than each edit in isolation: tracing diagnoses emerging interference, and editing executes controlled interventions whose mechanistic footprint is continuously audited.

Summary. These directions formalize a reciprocal contract: *tracing is actionable only when edit-validated*, and *editing is trustworthy only when mechanism-validated*. Closing this loop is key to scalable, reliable knowledge updates in models.

5 Conclusion

This work reveals a fundamental misalignment between knowledge tracing and knowledge editing in large language models. While tracing diagnoses internal knowledge, it often lacks causal leverage for effective intervention, and existing evaluations fail to capture structural degradation after editing. We call for a synergistic paradigm where tracing guides intervention and editing validates causal correctness, enabling more controllable models.

6 Limitation

This study examines the interplay between knowledge tracing and knowledge editing and discusses a closed-loop perspective between the two. Our analysis is biased toward how knowledge tracing can support editing, while the complementary role of editing in informing and refining tracing methods is not explored in depth. Moreover, our discussion focuses on transformer-based language models, extensions to alternative architectures such as Mamba and to multimodal settings are only briefly considered and remain open directions for future work.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online. Association for Computational Linguistics.

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Riccardo Ali, Francesco Caso, Christopher Irwin, and Pietro Liò. 2025. Entropy-lens: The information signature of transformer computations. *Preprint*, arXiv:2502.16570.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2018. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations*.

Dana Arad, Aaron Mueller, and Yonatan Belinkov. 2025. SAEs are good for steering – if you select the right features. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10252–10270, Suzhou, China. Association for Computational Linguistics.

Meghna P Ayyar, Jenny Benois-Pineau, and Akka Zemmari. 2025. There is more to attention: Statistical filtering enhances explanations in vision transformers. *arXiv preprint arXiv:2510.06070*.

Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. 2025. Steering large language model activations in sparse spaces. In *Second Conference on Language Modeling*.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

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

Baolong Bi, Shenghua Liu, Lingrui Mei, Yiwei Wang, Pengliang Ji, and Xueqi Cheng. 2024a. Decoding by contrasting knowledge: Enhancing llms' confidence on edited facts. *arXiv preprint arXiv:2405.11613*.

Baolong Bi, Shenghua Liu, Yiwei Wang, Lingrui Mei, Hongcheng Gao, Yilong Xu, and Xueqi Cheng. 2024b. Adaptive token biaser: Knowledge editing via biasing key entities. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11071–11083, Miami, Florida, USA. Association for Computational Linguistics.

Dan Braun, Jordan Taylor, Nicholas Goldowsky-Dill, and Lee Sharkey. 2024. Identifying functionally important features with end-to-end sparse dictionary learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Mingda Chen, Yang Li, Karthik Padthe, Rulin Shao, Alicia Yi Sun, Luke Zettlemoyer, Gargi Ghosh, and Wen-tau Yih. 2025a. Improving factuality with explicit working memory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11199–11213, Vienna, Austria. Association for Computational Linguistics.

Yuheng Chen, Pengfei Cao, Yubo Chen, Yining Wang, Shengping Liu, Kang Liu, and Jun Zhao. 2025b. Cracking factual knowledge: A comprehensive analysis of degenerate knowledge neurons in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10240–10261, Vienna, Austria. Association for Computational Linguistics.

Zhuo Chen, Yufeng Huang, Jiaoyan Chen, Yuxia Geng, Yin Fang, Jeff Z. Pan, Ningyu Zhang, and Wen Zhang. 2022. Lako: Knowledge-driven Visual Question Answering via Late Knowledge-to-text Injection. In *Proc. of IJCKG2022*, pages 20–29.

Zhendong Chu, Shen Wang, Jian Xie, Tinghui Zhu, Yibo Yan, Jinheng Ye, Aoxiao Zhong, Xuming Hu, Jing Liang, Philip S Yu, and 1 others. 2025. Llm agents for education: Advances and applications. *arXiv preprint arXiv:2503.11733*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

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

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. 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, Dublin, Ireland. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jianshuo Dong, Yutong Zhang, Liu Yan, Zhenyu Zhong, Tao Wei, Ke Xu, Minlie Huang, Chao Zhang, and Han Qiu. 2025. "I've decided to leak": Probing internals behind prompt leakage intents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21329–21359, Suzhou, China. Association for Computational Linguistics.

Yiming Du, Wenyu Huang, Danna Zheng, Zhaowei Wang, Sebastien Montella, Mirella Lapata, Kam-Fai Wong, and Jeff Z. Pan. 2025. Rethinking memory in llm based agents: Representations, operations, and emerging topics. *Preprint*, arXiv:2505.00675.

Clément Dumas, Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2025. Separating tongue from thought: Activation patching reveals language-agnostic concept representations in transformers. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31822–31841, Vienna, Austria. Association for Computational Linguistics.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. Alphaedit: Null-space constrained model editing for language models. In *The Thirteenth International Conference on Learning Representations*.

Thomas Fel, Ekdeep Singh Lubana, Jacob S. Prince, Matthew Kowal, Victor Boutin, Isabel Papadimitriou, Binxu Wang, Martin Wattenberg, Demba E. Ba, and Talia Konkle. 2025. Archetypal SAE: Adaptive and stable dictionary learning for concept extraction in large vision models. In *Forty-second International Conference on Machine Learning*.

Siddhartha Gairola, Moritz Böhle, Francesco Locatello, and Bernt Schiele. 2025. How to probe: Simple yet effective techniques for improving post-hoc explanations. In *The Thirteenth International Conference on Learning Representations*.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and 1 others. 2025. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26(83):1–64.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 9574–9586. Curran Associates, Inc.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. 2022. Inducing causal structure for interpretable neural networks. In *International Conference on Machine Learning*, pages 7324–7338. PMLR.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. 2024. Finding alignments between interpretable causal variables and distributed neural representations. In *Causal Learning and Reasoning*, pages 160–187. PMLR.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning LLMs on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7765–7784, Miami, Florida, USA. Association for Computational Linguistics.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 2024. Model editing harms general abilities of large language models: Regularization to the rescue. *arXiv preprint arXiv:2401.04700*.

Akshat Gupta, Dev Sajnani, and Gopala Anumanchipalli. 2024. A unified framework for model editing. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15403–15418, Miami, Florida, USA. Association for Computational Linguistics.

Tal Haklay, Hadas Orgad, David Bau, Aaron Mueller, and Yonatan Belinkov. 2025. Position-aware automatic circuit discovery. In *Proceedings of the 63rd Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 2792–2817, Vienna, Austria. Association for Computational Linguistics.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with GRACE: Lifelong model editing with discrete key-value adaptors. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Peter Hase, Thomas Hofweber, Xiang Zhou, Elias Stengel-Eskin, and Mohit Bansal. 2024. Fundamental problems with model editing: How should rational belief revision work in llms? *Preprint*, arXiv:2406.19354.

Guoxiu He, Xin Song, and Aixin Sun. 2025. Knowledge updating? no more model editing! just selective contextual reasoning. *arXiv preprint arXiv:2503.05212*.

Jie He, Víctor Gutiérrez-Basulto, and Jeff Z. Pan. 2023. BUCA: A Binary Classification Approach to Unsupervised Commonsense Question Answering. In *Proc. of ACL2023*.

Evan Hernandez, Belinda Z Li, and Jacob Andreas. 2023. Inspecting and editing knowledge representations in language models. *arXiv preprint arXiv:2304.00740*.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

Aliyah R. Hsu, Georgia Zhou, Yeshwanth Cherapanamjeri, Yaxuan Huang, Anobel Odisho, Peter R. Carroll, and Bin Yu. 2025. Efficient automated circuit discovery in transformers using contextual decomposition. In *The Thirteenth International Conference on Learning Representations*.

Nan Hu, Yike Wu, Guilin Qi, Dehai Min, Jiaoyan Chen, Jeff Z Pan, and Zafar Ali. 2023. An Empirical Study of Pre-trained Language Models in Simple Knowledge Graph Question Answering. In *Journal of World Wide Web*, pages 1–32.

Yaojie Hu and Jin Tian. 2022. Neuron dependency graphs: A causal abstraction of neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9020–9040. PMLR.

Baixiang Huang, Canyu Chen, Xiongxiao Xu, Ali Payani, and Kai Shu. 2024. Can knowledge editing really correct hallucinations? *arXiv preprint arXiv:2410.16251*.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-patcher: One mistake worth one neuron. In *The Eleventh International Conference on Learning Representations*.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Houcheng Jiang, Junfeng Fang, Ningyu Zhang, Guojun Ma, Mingyang Wan, Xiang Wang, Xiangnan He, and Tat-seng Chua. 2025. Anyedit: Edit any knowledge encoded in language models. *arXiv preprint arXiv:2502.05628*.

Hao Kang, Tevin Wang, and Chenyan Xiong. 2025. Interpret and control dense retrieval with sparse latent features. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 700–709, Albuquerque, New Mexico. Association for Computational Linguistics.

Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. 2021. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5050–5058.

Shahar Katz, Yonatan Belinkov, Mor Geva, and Lior Wolf. 2024. Backward lens: Projecting language model gradients into the vocabulary space. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2390–2422, Miami, Florida, USA. Association for Computational Linguistics.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and 1 others. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR.

Pang Wei W Koh, Kai-Siang Ang, Hubert Teo, and Percy S Liang. 2019. On the accuracy of influence functions for measuring group effects. *Advances in neural information processing systems*, 32.

Kai Konen, Sophie Jentzsch, Diaoulé Diallo, Peer Schütt, Oliver Bensch, Roxanne El Baff, Dominik

Opitz, and Tobias Hecking. 2024. Style vectors for steering generative large language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 782–802, St. Julian's, Malta. Association for Computational Linguistics.

Michael Lan, Philip Torr, and Fazl Barez. 2024. Towards interpretable sequence continuation: Analyzing shared circuits in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12576–12601, Miami, Florida, USA. Association for Computational Linguistics.

Anna Langedijk, Hosein Mohebbi, Gabriele Sarti, Willem Zuidema, and Jaap Jumelet. 2024. DecoderLens: Layerwise interpretation of encoder-decoder transformers. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4764–4780, Mexico City, Mexico. Association for Computational Linguistics.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. 2025. Programming refusal with conditional activation steering. In *The Thirteenth International Conference on Learning Representations*.

Jiaang Li, Quan Wang, Zhongnan Wang, Yongdong Zhang, and Zhendong Mao. 2025a. Elder: Enhancing lifelong model editing with mixture-of-lora. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24440–24448.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530.

Qi Li, Xiang Liu, Zhenheng Tang, Peijie Dong, Zeyu Li, Xinglin Pan, and Xiaowen Chu. 2024a. Should we really edit language models? on the evaluation of edited language models. *Advances in Neural Information Processing Systems*, 37:30850–30885.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024b. Pmet: Precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18564–18572.

Yanhong Li, Chunling Fan, Mingqing Huang, and Chengming Li. 2024c. Learning from mistakes: A comprehensive review of knowledge editing for large language models. In *2024 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pages 563–569.

Zhe Li, Wei Zhao, Yige Li, and Jun Sun. 2025b. Do influence functions work on large language models? In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 14367–14382, Suzhou, China. Association for Computational Linguistics.

Yibing Liu, Haoliang Li, Yangyang Guo, Chenqi Kong, Jing Li, and Shiqi Wang. 2022. Rethinking attention-model explainability through faithfulness violation test. In *International conference on machine learning*, pages 13807–13824. PMLR.

Daniel Lundstrom and Meisam Razaviyayn. 2025. Four axiomatic characterizations of the integrated gradients attribution method. *Journal of Machine Learning Research*, 26(177):1–31.

Vittorio Mazzia, Alessandro Pedrani, Andrea Caciolai, Kay Rottmann, and Davide Bernardi. 2025. A survey on knowledge editing of neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 36(7):11759–11775.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2024. Circuit component reuse across tasks in transformer language models. In *The Twelfth International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022a. Fast model editing at scale. In *International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15817–15831. PMLR.

Jesse Mu and Jacob Andreas. 2020. Compositional explanations of neurons. *Advances in Neural Information Processing Systems*, 33:17153–17163.

Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. 2025. Towards interpreting visual information processing in vision-language models. In *The Thirteenth International Conference on Learning Representations*.

Ian E Nielsen, Dimah Dera, Ghulam Rasool, Ravi P Ramachandran, and Nidhal Carla Bouaynaya. 2022. Robust explainability: A tutorial on gradient-based attribution methods for deep neural networks. *IEEE Signal Processing Magazine*, 39(4):73–84.

Nostalgebraist. 2020. interpreting gpt: the logit lens.

Jeff Z. Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo

Lissandrini, ussa Biswas, Gerard de Melo, Angela Bonifati, Edlira Vakaj, Mauro Dragoni, and amien Graux. 2023. Large language models and knowledge graphs: Opportunities and challenges. *Transactions on Graph Data and Knowledge*.

Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. 2021. A plug-and-play method for controlled text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3973–3997, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. 2020. Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4782–4793, Online. Association for Computational Linguistics.

Siyuan Qi, Bangcheng Yang, Kailin Jiang, Xiaobo Wang, Jiaqi Li, Yifan Zhong, Yaodong Yang, and Zilong Zheng. 2025. In-context editing: Learning knowledge from self-induced distributions. In *The Thirteenth International Conference on Learning Representations*.

Shanbao Qiao, Xuebing Liu, and Seung-Hoon Na. 2024. COMEM: In-context retrieval-augmented mass-editing memory in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2333–2347, Mexico City, Mexico. Association for Computational Linguistics.

Yifu QIU, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo Ponti, and Shay B Cohen. 2024. Spectral editing of activations for large language model alignment. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Shuanghong Shen, Qi Liu, Zhenya Huang, Yonghe Zheng, Minghao Yin, Minjuan Wang, and Enhong Chen. 2024. A survey of knowledge tracing: Models, variants, and applications. *IEEE Transactions on Learning Technologies*, 17:1858–1879.

Wei Shi, Sihang Li, Tao Liang, Mingyang Wan, Guojun Ma, Xiang Wang, and Xiangnan He. 2025a. Route sparse autoencoder to interpret large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6812–6826, Suzhou, China. Association for Computational Linguistics.

Xiang Shi, Jiawei Liu, Yinpeng Liu, Qikai Cheng, and Wei Lu. 2025b. Know where to go: Make llm a relevant, responsible, and trustworthy searchers. *Decision Support Systems*, 188:114354.

Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. 2024. Rethinking interpretability in the era of large language models. *arXiv preprint arXiv:2402.01761*.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. 2025. Improving instruction-following in language models through activation steering. In *The Thirteenth International Conference on Learning Representations*.

Nishant Subramani, Nivedita Suresh, and Matthew E Peters. 2022. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 566–581.

Zengkui Sun, Yijin Liu, Jiaan Wang, Fandong Meng, Jinan Xu, Yufeng Chen, and Jie Zhou. 2024. Outdated issue aware decoding for factual knowledge editing. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9282–9293, Bangkok, Thailand. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Denis Sutter, Julian Minder, Thomas Hofmann, and Tiago Pimentel. 2025. The non-linear representation dilemma: Is causal abstraction enough for mechanistic interpretability? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Ryota Takatsuki, Sonia Joseph, Ippei Fujisawa, and Ryota Kanai. 2025. Decoding vision transformers: the diffusion steering lens. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4819–4824.

Chenmien Tan, Ge Zhang, and Jie Fu. 2024. Massive editing for large language models via meta learning. In *The Twelfth International Conference on Learning Representations*.

Juanhe (TJ) Tan. 2023. Causal abstraction for chain-of-thought reasoning in arithmetic word problems. In *Proceedings of the 6th BlackboxNLP Workshop:*

Analyzing and Interpreting Neural Networks for NLP, pages 155–168, Singapore. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Lukas Thede, Karsten Roth, Matthias Bethge, Zeynep Akata, and Thomas Hartvigsen. 2025. Wikibigedit: Understanding the limits of lifelong knowledge editing in LLMs. In *Forty-second International Conference on Machine Learning*.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. 2024. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*.

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

Ke Wang, Yiming QIN, Nikolaos Dimitriadis, Alessandro Favero, and Pascal Frossard. 2025a. MEMOIR: Lifelong model editing with minimal overwrite and informed retention for LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2024a. WISE: Rethinking the knowledge memory for lifelong model editing of large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Pinzheng Wang, Zecheng Tang, Keyan Zhou, Juntao Li, Qiaoming Zhu, and Min Zhang. 2025b. Revealing and mitigating over-attention in knowledge editing. In *The Thirteenth International Conference on Learning Representations*.

Renzhi Wang and Piji Li. 2024. LEMoE: Advanced mixture of experts adaptor for lifelong model editing of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2551–2575, Miami, Florida, USA. Association for Computational Linguistics.

Shiqi Wang, Qi Wang, Runliang Niu, He Kong, and Yi Chang. 2025c. MicroEdit: Neuron-level knowledge disentanglement and localization in lifelong model editing. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33870–33884, Suzhou, China. Association for Computational Linguistics.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2024b. Knowledge editing for large language models: A survey. *ACM Computing Surveys*, 57(3):1–37.

Weixuan Wang, Barry Haddow, and Alexandra Birch. 2024c. Retrieval-augmented multilingual knowledge editing. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 335–354, Bangkok, Thailand. Association for Computational Linguistics.

Wenxuan Wang, Zizhan Ma, Zheng Wang, Chenghan Wu, Jiaming Ji, Wenting Chen, Xiang Li, and Yixuan Yuan. 2025d. A survey of llm-based agents in medicine: How far are we from baymax? *arXiv preprint arXiv:2502.11211*.

Yifei Wang, Yuheng Chen, Wanting Wen, Yu Sheng, Linjing Li, and Daniel Dajun Zeng. 2024d. Unveiling factual recall behaviors of large language models through knowledge neurons. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7388–7402, Miami, Florida, USA. Association for Computational Linguistics.

Yiwei Wang, Muhao Chen, Nanyun Peng, and Kai-Wei Chang. 2024e. Deepedit: Knowledge editing as decoding with constraints. *arXiv preprint arXiv:2401.10471*.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. 2023. Interpretability at scale: Identifying causal mechanisms in alpaca. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Jianhao Yan, Futing Wang, Yun Luo, Yafu Li, and Yue Zhang. 2025. Keys to robust edits: from theoretical insights to practical advances. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22545–22560.

Yunzhi Yao, Jizhan Fang, Jia-Chen Gu, Ningyu Zhang, Shumin Deng, Huajun Chen, and Nanyun Peng. 2025. CaKE: Circuit-aware editing enables generalizable knowledge learners. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 11377–11393, Suzhou, China. Association for Computational Linguistics.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. *Preprint*, arXiv:2305.13172.

Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and Huajun Chen. 2024. Knowledge circuits in pretrained transformers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. 2024. Melo: enhancing model editing with neuron-indexed dynamic lora. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Fred Zhang and Neel Nanda. 2024. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*.

Mengqi Zhang, Xiaotian Ye, Qiang Liu, Pengjie Ren, Shu Wu, and Zhumin Chen. 2024a. Uncovering overfitting in large language model editing. *arXiv preprint arXiv:2410.07819*.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, and 3 others. 2024b. A comprehensive study of knowledge editing for large language models. *Preprint*, arXiv:2401.01286.

Zheyuan Zhang, Daniel Zhang-Li, Jifan Yu, Linlu Gong, Jinchang Zhou, Zhanxin Hao, Jianxiao Jiang, Jie Cao, Huiqin Liu, Zhiyuan Liu, and 1 others. 2025. Simulating classroom education with llm-empowered agents. 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 10364–10379.

Runcong Zhao, Chengyu Cao, Qinglin Zhu, Xiucheng Ly, Shun Shao, Lin Gui, Ruifeng Xu, and Yulan He. 2025. Sparse activation editing for reliable instruction following in narratives. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25828–25843, Suzhou, China. Association for Computational Linguistics.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4862–4876, Singapore. Association for Computational Linguistics.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. 2024. Mquake: Assessing knowledge editing in language models via multi-hop questions. *Preprint*, arXiv:2305.14795.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.

Hanlun Zhu, Yunshi Lan, Xiang Li, and Weining Qian. 2025. Initializing and retrofitting key-value adaptors for traceable model editing. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2958–2971, Vienna, Austria. Association for Computational Linguistics.

A Paper collection

We note that Knowledge Tracing traditionally refers to a student modeling task in online education, aiming to infer learners' latent knowledge states from behavioral sequences. (Shen et al., 2024) In contrast, recent LLM studies employ the term to describe a fundamentally different problem: tracing the internal representations and causal pathways through which factual knowledge influences model predictions.

In this work, we use Knowledge Tracing in a different sense, referring to methods that trace where and how factual knowledge functionally participates in the internal computation of large language models.

To got the related paper, we use the above steps:

- We use the Tools [2] to collect papers from past 3 years (2022 to 2025) from Top NLP and ML conferences (i.e., ACL, EMNLP, NeurIPS, ICML, ICLR). Got 50268 paper in total.
- we use nougat [3] to extract the paper's abstract and title information from the pdf file. And use all-MiniLM-L6-v2 [4] to filter the relevant paper with abstract and title embedding. We got 24672 paper.
- To reduce the bias from different research area, we use GPT4o to score the relevance of a paper (1-10) with the keywords such as 'Interpretable', 'Controlling output by modifying model internals', 'Model/knowledge Editing' and 'Causal tracing and intervention in deep models'. And got the paper that relevance score > 6. We got 1980 paper.
- We then collect their publication date and citation numbers from Semantic Scholar API[5] and then calculate the RCI for each paper, and choose the paper that RCI score > 1. We got 355 paper.
- And then use A scholarly research assistant from Ai2 [6] to collect the most related paper to knowledge Tracing and Editing (ignore the year with this tools.). Then we compare the results with the 355 paper to filter the noise and add the ignore paper. We got 220 paper with high related topic, contain 130 Tracing paper and 90 Editing paper. And recalculate the RCI score based on these papers.

[2] https://github.com/hegongshan/paper-downloader

[3] https://facebookresearch.github.io/nougat/

[4] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[5] https://www.semanticscholar.org/product/api

[6] https://asta.allen.ai/

B Tools for Editing and Tracing

Model editing has benefited from a rapid standardization of its workflow, supported by comprehensive toolkits like *EasyEdit* and *FastEdit*. This robust infrastructure significantly lowers the entry barrier for developing and scaling efficient intervention methods. In contrast, the tracing ecosystem—while featuring influential tools like *TransformerLens* and *Circuit-tracer*, remains heavily constrained by its dependency on low-level architectural interventions (e.g., neuron-level hooking). As shown in Table 4, tracing-related tools are numerous and diverse, covering visualization (e.g., BertViz, Transformer-Explainer), attribution (e.g., Captum, SHAP), circuit analysis (e.g., circuit-tracer), and interactive analysis platforms (e.g., LIT).

These methods often lack cross-model portability, making it difficult to generalize tracing conclusions to edited models or heterogeneous architectures.

C Paper Scores

In this work, we use Relative Citation Index (RCI) metric(Du et al., 2025), which estimate the expected citations with respect to publication age to prevent bias between original citations from different publication dates. The age A_i of a paper p_i is computed as:

$$A = T - Year_i, \tag{1}$$

where T is the date when the citation is collected (20th Nov. 2025) and $Year_i$ is the year where paper i is first published. Thus, we can model the relation between citation number C_i and age A_i of paper p_i in three different way, which are:

linear model:

$$C_i = \beta + \alpha A_i \tag{2}$$

exponential model:

$$C_i = \exp(\beta + \alpha A_i) \tag{3}$$

log-log regression model:

$$\log(C_i + 1) = \beta + \alpha \log A_i + \epsilon_i \tag{4}$$

Tools	Stars	Link
transformerlen	2.9k	https://github.com/TransformerLensOrg/TransformerLen
BertViz	7.9k	https://github.com/jessevig/bertviz
LIT	3.6k	https://github.com/PAIR-code/lit
ECCO	2.1k	https://github.com/jalammar/ecco
Captum	5.5k	https://github.com/pytorch/captum
Transformers-Interpret	1.4k	https://github.com/cdpierse/transformers-interpret
AllenNLP Interpret	11.9k	https://github.com/allenai/allennlp-interpret
transformer-explainer	6.3k	https://github.com/poloclub/transformer-explainer
SHAP	24.9k	https://github.com/slundberg/shap
transformer-debugger	4.1k	https://github.com/openai/transformer-debugger
llm-transparency-tool Public	1.2k	https://github.com/facebookresearch/llm-transparency-tool
circuit-tracer	2.5k	https://github.com/safety-research/circuit-tracer/tree/main
bi-att-flow	1.5k	https://github.com/allenai/bi-att-flow
EasyEdit	2.7k	https://github.com/zjunlp/EasyEdit/tree/main
FastEdit	1.4k	https://github.com/hiyouga/FastEdit

Table 4: Popular Tools for knowledge Editing and Tracing.

We pick log-log regression model to compute the expected citation for next step[7], and we are able to obtain the expected citation number $\hat{C}_i$ of paper p_i with age A_i as:

$$\hat{C}_i = \exp(\hat{\beta}) A_i^{\hat{\alpha}} \tag{5}$$

Then we compute the relative citation index RCI_i of paper p_i as:

$$RCI_i = \frac{C_i}{\hat{C}_i} \tag{6}$$

When $RCI_i >= 1$, we consider this paper over-cited than its expectations, and vice versa. In this paper, we focus on the paper with $RCI >= 1$, for which we believe has more influence.

The Citation Velocity is calculate by:

$$V = \frac{C_i}{A}, \tag{7}$$

captures its current research momentum.

The Important score is calculate by:

$$Importance_i = 2 \times \frac{RCI_i \times \overline{Velocity}_i}{RCI_i + \overline{Velocity}_i} \tag{8}$$

Where the Normalized Citation Velocity is :

$$\overline{Velocity}_i = \frac{Velocity_i}{\frac{1}{N}\sum j = 1^N Velocity_j} \tag{9}$$

As formulated in Eq. 8, this score is the harmonic mean of the *Relative Citation Index* (RCI) and the *Normalized Citation Velocity*.

[7]The estimation is: $\hat{\beta} = 1.815$, $\hat{\alpha} = 0.770$

D Relative Citation Index and Citation Velocity

D.1 Quadrant analysis

The Top Paper in Figure 3b for each type is:

P198: "Why Should I Trust You?": Explaining the Predictions of Any Classifier (Ribeiro et al., 2016)

P095: Investigating Gender Bias in Language Models Using Causal Mediation Analysis (Vig et al., 2020)

P016: Axiomatic Attribution for Deep Networks (Sundararajan et al., 2017)

P213: AnyEdit: Edit Any Knowledge Encoded in Language Models (Jiang et al., 2025)

P113: Locating and Editing Factual Associations in GPT (Meng et al., 2022)

P114: Mass-Editing Memory in a Transformer (Meng et al., 2023)

For the High-impact & High-momentum quadrant (Q1): Tracing papers overwhelmingly dominate the "Mainstream Core" with 59 works compared to 29 in Editing. This nearly 2:1 ratio indicates that interpretability research has not only established a solid foundation but continues to drive the field's primary intellectual momentum. For the most important paper in Q1 also shows the importance for tracing paper is higher than the editing paper.

For the Emerging & High-momentum quadrant (Q2): The bottleneck for Editing is most evident in

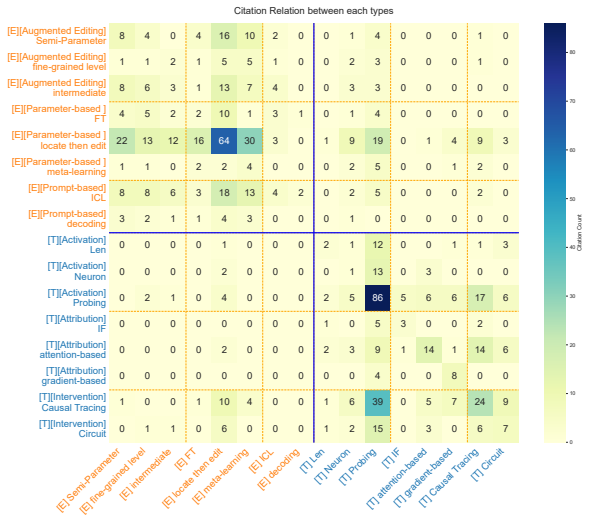


Figure 4: Cross-category citation analysis

this quadrant. While 16 Tracing papers are categorized as high-momentum emerging works, only 3 Editing papers achieve this status. This suggests that despite the surge in publication volume, Editing struggles to produce "breakout" research that captures immediate and widespread academic attention.

For the Low-impact & Low-momentum quadrant (Q3): Both fields have a significant presence here (38 Editing vs. 49 Tracing), representing the "long tail" of incremental research. However, for Editing, this quadrant accounts for the largest share of its total publications, reflecting a high volume of work that has yet to achieve significant resonance.

For the Established & High Influence quadrant (Q4): Notably, Editing has more "Legacy" papers in Q4 (12) compared to Tracing (7). This indicates that while early Editing methods achieved high citation counts, the field has struggled to maintain that same level of influence in more recent, high-velocity cycles compared to the sustained impact of Tracing.

D.2 Cross-category citation

Cross-category citation (Figure 4) reveals a strongly asymmetric dependency between knowledge editing methods and interpretability-oriented tracing techniques. This hierarchy positions interpretability primarily as a supporting instrument for localization, rather than a co-evolving research objective.

Editing approaches exhibit substantial citation flows toward foundational tracing methods, yet the reverse is rarely true. Specifically, the Locate-then-edit paradigm acts as the primary consumer of interpretability research, citing Probing (19 citations), Causal Tracing (9 citations), and Circuit analysis (3 citations). Similarly, Semi-parameter and Intermediate editing methods consistently reference Probing (4 and 3 citations respectively) to validate their intervention targets. In stark contrast, tracing show minimal reverse dependence, for instance, Causal Tracing and Circuit-based works cite the entire spectrum of editing methods fewer than 10 times in total.

Locate-then-edit functions as a structural nexus, maintaining dense cross-citations that bridge heterogeneous editing strategies. It exhibits strong connectivity with Semi-parameter (22 citations), Fine-grained (13 citations), Intermediate (12 citations), FT (16 citations), and ICL-based editing (18 citations). This central positioning confirms that "localization" remains the dominant mental model for knowledge manipulation, even if the localized components do not always provide the necessary controllability for stable editing. *Prompt-based and Augmented Editing methodologies exhibit a superficial and unidirectional reliance on interpretability foundations.* Compared to direct parameter-based interventions, these approaches maintain a marginal dependency on tracing research.

Table 5 summarizes the alignment between knowledge editing methods and tracing types. Parameter-Based Editing methods, such as ROME, MEMIT, and AlphaEdit, consistently leverage intervention- or attribution-based tracing to guide fine-grained parameter updates, reflecting a strong dependency on tracing signals. In contrast, Augmented and Prompt-Based Editing methods show limited or implicit reliance on tracing, often operating at the module, hidden state, or token level without explicit mechanistic guidance. This disparity suggests that while tracing could provide actionable guidance for these latter categories, current research has largely overlooked this potential, highlighting an opportunity for future work to more systematically integrate tracing into diverse editing paradigms.

E causal diagnostic (AlphaEdit + causal tracing)

Full Causal trace visualization results are provided in Figure 5, 6, 7, 8, and 9.

Editing Method	Tracing Alignment	Interventional Module	Edit Granularity
Parameter-Based Editing			
ROME (Meng et al., 2022)	Intervention-based	MLP block	Parameters level
MEMIT (Meng et al., 2023)	Intervention-based	MLP block	Parameters level
AlphaEdit (Fang et al., 2025)	Intervention-based	MLP block	Parameters level
MEND (Mitchell et al., 2022a)	Attribution-based	MLP block	Parameters level
KN (Dai et al., 2022)	Activation-based	Neurons	Neurons level
CAKE (Yao et al., 2025)	Intervention-based	MLP block	Parameters level
ICE (Qi et al., 2025)	-	Hidden states	Parameters level
MALMEN (Tan et al., 2024)	*Attribution-based*	Parameters	Parameters level
LTI (Zhang et al., 2024a)	-	MLP block	Parameters level
Augmented Editing			
SERAC (Mitchell et al., 2022b)	-	Module	Token level
WISE (Wang et al., 2024a)	-	MLP block	Hidden states level
T-Patch (Huang et al., 2023)	-	Hidden states	Neurons level
GRACE (Hartvigsen et al., 2023)	-	Hidden states	Hidden states level
MELO (Yu et al., 2024)	-	Module	Hidden states level
REMEDI (Hernandez et al., 2023)	*Attribution-based*	Hidden states	Hidden states level
Prompt-Based Editing			
IKE (Zheng et al., 2023)	-	Token	Token level
DeCK (Bi et al., 2024a)	-	Token	Token level
ATBIAS (Bi et al., 2024b)	*Activation-based*	Logits length	Token level
DISCO (Sun et al., 2024)	*Activation-based*	Logits length	Token level
MeLLo (Zhong et al., 2024)	-	Token	Token level

Table 5: Analysis of misalignment between knowledge editing and tracing. We categorize results into **Parameter-Based Editing**, **Augmented Editing**, and **Prompt-Based Editing**. Normal text represents explicit alignment, *italic text* represents implicit alignment, and '-' denotes total misalignment.

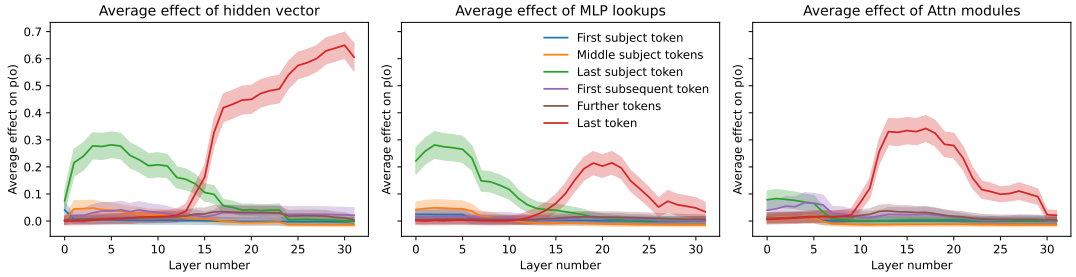


Figure 5: Before Editing, Causal trace on 100 Matched-Data (MD)

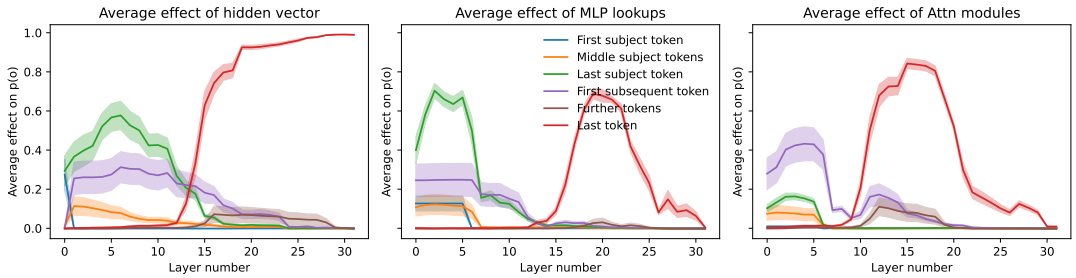


Figure 6: Post-Edit model (Edit on 100 Matched-Data (MD))

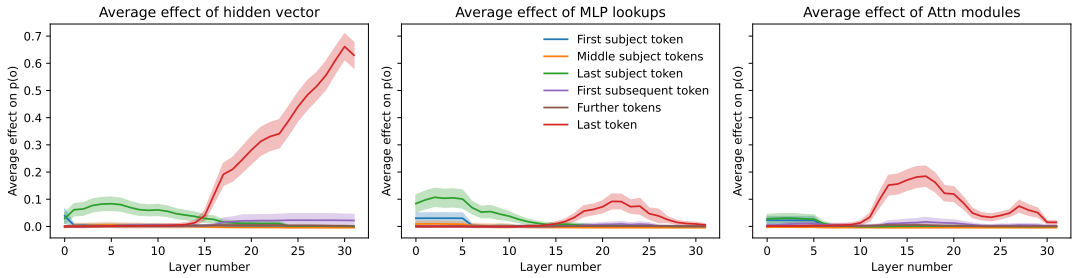


Figure 7: Post-Edit model (Edit on 100 Independent-Data (ID))

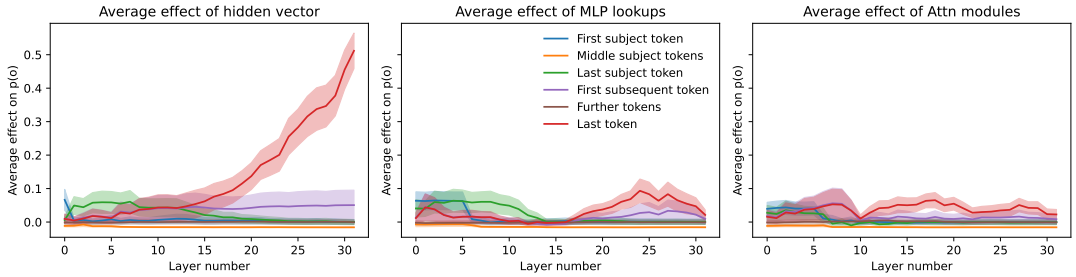


Figure 8: Post-Edit model (Edit on 300 Independent-Data (ID))

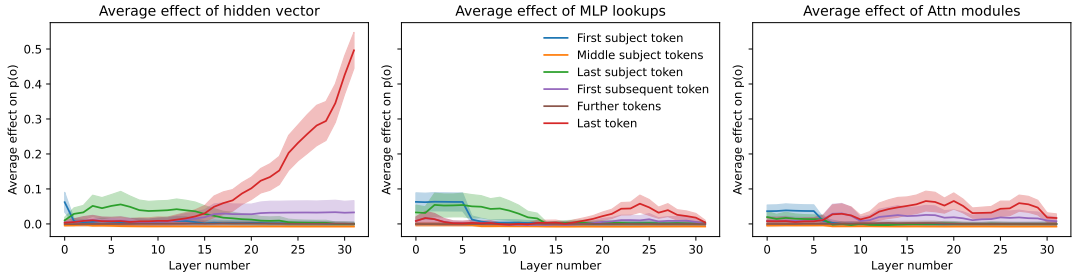


Figure 9: Post-Edit model (Edit on 500 Independent-Data (ID))

F Architectural landscape of KT and KE

The architectural landscape of the knowledge tracing and editing is provided in Figure 10.

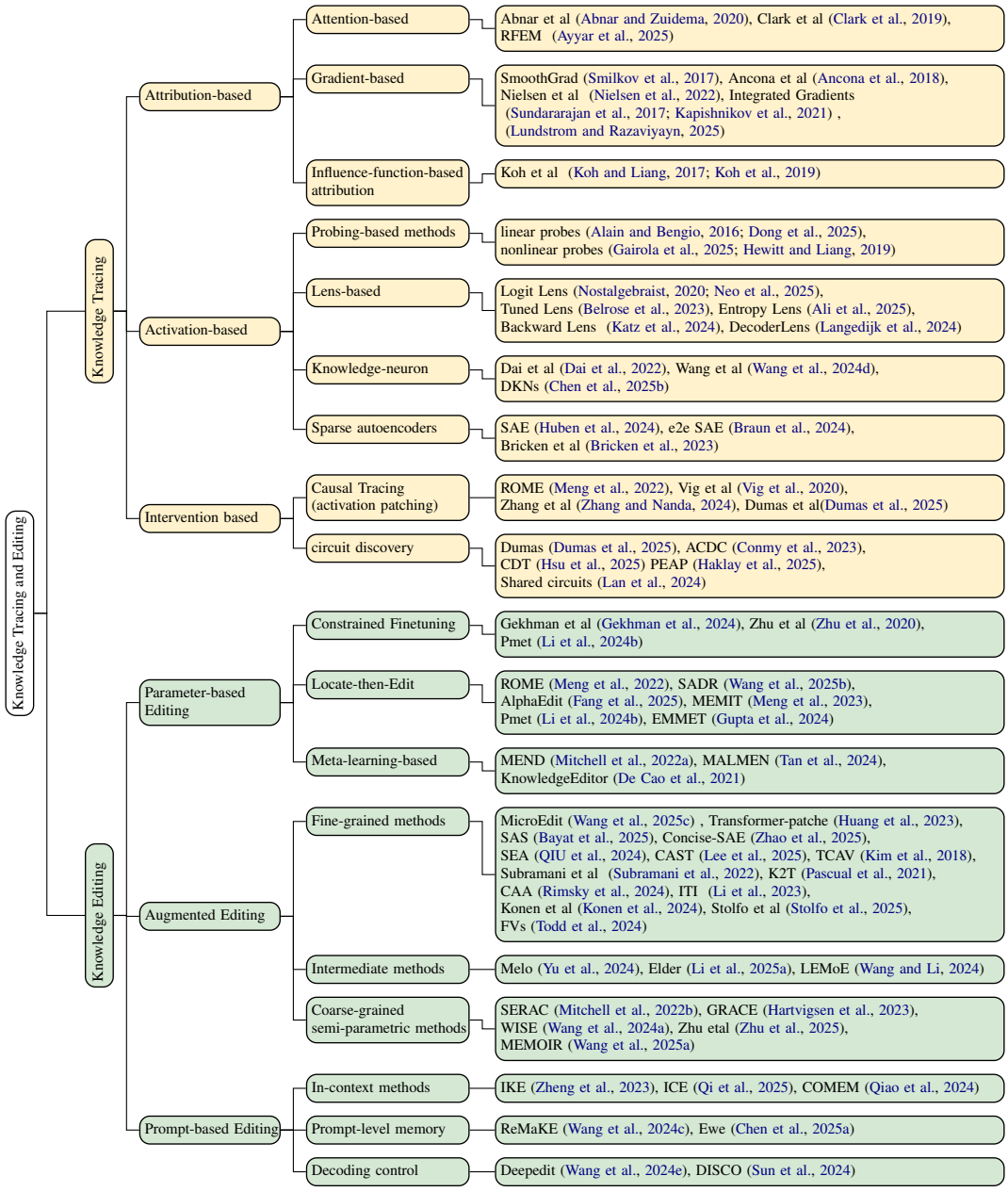


Figure 10: The overview of Knowledge tracing and editing methods