HYBRID-POLICY SELF-EDITING FOR COMPOSABLE UNSTRUCTURED KNOWLEDGE EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained large language models (LLMs) achieve remarkable performance across natural language tasks, yet their internal knowledge quickly becomes outdated in a fast-changing world. This motivates knowledge editing (KE), which updates specific knowledge in an LLM without changing unrelated knowledge. Recent works move from structured knowledge triples toward unstructured KE (UKE), where the edit is a free-form passage that may state multiple facts at once. Nonetheless, existing editors inject the passage but fail to *use* it: the edited model can recall the passage, but can neither answer atomic questions about its facts nor compose them into multi-hop reasoning. We attribute this missing property, which we term *composability*, to editors' passive reliance on the fixed passage as the sole learning source. In response, we cast editing as a proactive self-distillation from a privileged in-context state of the same model, which requires no external supervision. We further reveal that because the injected knowledge is novel, the pre-edited model's own rollouts rarely cover it, limiting the effectiveness of pure on-policy distillation. To close this gap, we propose HPSE, which builds a hybrid rollout that steps in to place missing facts onto the student's own trajectory precisely where its coverage fails, while staying on-policy elsewhere. We theoretically analyze HPSE's advantage over pure on-policy distillation, and empirically establish its plug-and-play improvements across four LLM backbones and two KE editors under various scenarios. We release our datasets and evaluation code at https://anonymous.4open.science/r/hpse-review-C778.

1 INTRODUCTION

Large language models (LLMs) demonstrate strong generalization across various language tasks (Vaswani et al., 2017; Brown et al., 2020; Raffel et al., 2020; Bommasani et al., 2021; Zhou et al., 2023). As model sizes grow, LLMs further exhibit emergent abilities to follow natural language instructions (Ouyang et al., 2022; Dong et al., 2024) and zero-shot adaptability to unseen tasks (Kojima et al., 2022; Bubeck et al., 2023), forming a new foundation for modern NLP.

However, real-world LLM deployment remains far from resolved unless LLMs can continually refresh and internalize new knowledge beyond their training data. In a fast-changing world, an LLM's *static* knowledge quickly goes out of date, leading to factual errors or even unsafe generations (De Cao et al., 2021; Hartvigsen et al., 2023). Yet retraining the model to absorb each update is prohibitively expensive. As a remedy, *Knowledge Editing* (KE) has been proposed to address this by updating an LLM with a *specific* piece of knowledge (Wang et al., 2023; Zhang et al., 2024c).

Given a direct, well-structured statement of new knowledge, e.g., a knowledge triple naming the new owner of a recently acquired company, KE typically seeks *selective* parameter updates that keep unrelated knowledge and general capabilities intact. To name a few, Meng et al. (2023); Fang et al. (2025) apply closed-form updates to a few causally traced MLP layers that serve as knowledge storage (Dai et al., 2021), Wang et al. (2024a) directly learn sparse low-rank updates. Liu et al. (2025b) steer selected representations with non-linear updates. Recent works move towards the more realistic setting of *unstructured* KE (UKE), where the new knowledge spans multiple aspects in free-form text. For example, an actual acquisition announcement is more likely to be a free-form passage that states multiple entangled facts at once, such as the new parent company, the incoming CEO, and the reshaped reporting structure, with no specification of which facts a user may later

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

query. UKE pursues this more challenging objective: treating the announcement passage itself as the knowledge to edit (Wu et al., 2024a). To this end, Deng et al. (2024) extend the edit from a few MLP layers to entire transformer blocks for larger capacity. Jiang et al. (2025) decompose the passage into sequential chunks for iterative editing; and Su et al. (2025) preserve the dependencies among chunk-wise updates. Together, these fine-grained designs make KE *precise*.

Nonetheless, the edited model loses two useful capabilities acquired through pretraining. The first is *decomposition*: after internalizing a long-form edit, the model can at best recall the whole passage yet cannot answer targeted questions about the individual facts it encodes (Zhou et al., 2026; Wang et al., 2026). The second is *composition*: having learned individual facts, the model cannot compose them for multi-hop reasoning (Zhong et al., 2023; 2025). Both phenomena have been analyzed in the literature, but mostly in isolation under separate scenarios (Zhang et al., 2024a; Yao et al., 2025), although they co-occur and are amplified in UKE. We use *composability* to denote whether edited knowledge, like its pretrained counterpart, can function as atomic facts that are flexibly extracted and recombined, and study decomposition and composition from this unified perspective.

In particular, we introduce an *untargeted* regime for UKE: a free-form knowledge statement is paired with a generic editing prompt that does not specify which facts are updated, mirroring practical requests where annotators offer only a brief summary of the new passage. We recast two representative benchmarks into a decomposition probe (Deng et al., 2024) and a composition probe (Zhong et al., 2025), respectively. We find that the evaluated methods, despite their different mechanisms, consistently fall short of composability under our regime. We attribute this failure to their *passive* reliance on the provided editing context as the sole learning source, which induces severe memorization and poor generalization (Zhang et al., 2024a; Liu et al., 2025a; Chu et al., 2025); existing augmentation-based remedies (Zhou et al., 2026; Yao et al., 2025; Wang et al., 2026) alleviate but do not eliminate this reliance. In summary, composability remains an open challenge, necessitating a generalizable UKE method; Sec. 2 details these findings. *Our untargeted regime and formulation of composability offer a unified view of what UKE requires in the wild. Together with our benchmarking protocol, these constitute the first key contribution of this work.*

Given the pivotal role of the editing context, we pursue a *proactive* strategy through on-policy self-distillation (OPSD): the edited model generates its own response trajectories, while a *privileged* state of the same model (one that reads the new passage in context) provides token-level distillation targets (Lu & Lab, 2025; Song & Zheng, 2026; Zhao et al., 2026), which has proved effective for adapting behavioral patterns in post-training (Chu et al., 2025; Lu & Lab, 2025; Song & Zheng, 2026). However, we reveal a *coverage failure* of OPSD in UKE: because the injected knowledge is novel, the model's own rollouts often go off-topic, yielding limited corrective supervision.

We propose HPSE, which lets the privileged model not only grade the edited model's responses, but also *step in* when a rollout deviates too far from a useful trajectory. This strategy forms hybrid, rather than purely on-policy, rollouts with better coverage, offering stronger supervision for UKE. We further support HPSE with theoretical justification. Notably, HPSE is a data-centric approach: it replaces only the training signal, making no assumption about which parameters are updated or how the update is parameterized, thereby offering plug-and-play flexibility across gradient-based KE editors. *The proposed HPSE and our theoretical analysis form another key contribution of this work.*

Our paper is organized as follows. Secs. 2 and 3 detail composable UKE and our HPSE, respectively. Sec. 4 evaluates HPSE extensively. Sec. 5 reviews related work, and Sec. 6 concludes the paper.

2 Composability Challenge in Unstructured Knowledge Editing

We formalize composability, an overlooked requirement in unstructured knowledge editing (UKE).

2.1 Preliminaries

For a text $\boldsymbol{x} = (x_1, \ldots, x_n)$, an LLM with parameters θ factorizes its likelihood as:

$$\pi_\theta(\boldsymbol{x}) = \prod_{i=1}^{n} \pi_\theta(x_i \mid x_1, \ldots, x_{i-1}) \triangleq \prod_{i=1}^{n} \pi_\theta(x_i \mid \boldsymbol{x}_{<i}),$$

where $\pi_\theta(x_i \mid \boldsymbol{x}_{<i})$ is the predicted distribution of token x_i given the previous $\boldsymbol{x}_{<i}$. The LLM is usually trained with maximum likelihood estimation (Hochreiter & Schmidhuber, 1997; Bengio et al., 2000; Sutskever et al., 2014; Cho et al., 2014; Yenduri et al., 2023). To generate a sentence $\boldsymbol{x}$, the LLM computes $\pi_\theta(x_i \mid \boldsymbol{x}_{<i})$ and draws x_i from it; then x_i is combined with $\boldsymbol{x}_{<i}$ as new inputs for future steps. Generation stops at an end-of-sequence token or the length limit.

Knowledge Editing (KE) (Wang et al., 2023; Zhang et al., 2024c) aims to update a pre-trained LLM's specific knowledge while preserving other knowledge unrelated to the update (a *locality* requirement). Structured KE expresses a piece of knowledge as a triple *(subject, relation, object)* and requires updating *object* to a new value: Given a natural-language pair $(\boldsymbol{x}, \boldsymbol{y})$ where the editing prompt $\boldsymbol{x}$ describes a *subject* and *relation*, KE asks the LLM to respond in natural language to $\boldsymbol{x}$ with the edited object $\boldsymbol{y}$. Building on KE, **Unstructured KE (UKE)** studies a more realistic regime, where a free-form passage $\boldsymbol{c}$ encodes *one* or *multiple* pieces of new knowledge, with no labeled knowledge triple, making it more challenging (Wu et al., 2024a; Yang et al., 2025; Jiang et al., 2025).

2.2 Formulating and Benchmarking UKE Composability

Composability Definition. KE is useful only if the knowledge it injects can be *used* by the LLM, not merely *memorized* in a way that can only be repeated under the exact query used for editing (Zhong et al., 2023; Deng et al., 2024; Zhang et al., 2024a; Zhong et al., 2025; Zhou et al., 2026). Crucially, when it comes to UKE, where multiple pieces of knowledge are injected at once, a *useful* edit entails the following two aspects, which we collectively refer to as *composability*.

The first aspect is *Decomposition.* Multiple pieces of knowledge are often encoded in the passage $\boldsymbol{c}$ and are injected jointly by editing the LLM with $\boldsymbol{c}$ directly (Deng et al., 2024). In this case, an ideal LLM should be able to decompose these facts and absorb them individually. After editing, any query about an atomic fact in the passage should be answered with the fact directly, rather than by repeating the whole passage (Zhou et al., 2026; Wang et al., 2026). The second aspect is *Composition,* a capability tied to *portability*, that requires related new facts to support multi-hop reasoning (Zhong et al., 2023; Wang et al., 2024c; Zhong et al., 2025).

We further consider how the new knowledge is presented for editing. In practice, human labelers may want to only summarize the passage briefly, instead of detailing which contents constitute the *updated* knowledge. To reflect this demand, we formulate UKE in an *untargeted regime:* the passage $\boldsymbol{c}$ is paired with a generic editing instruction like "`introduce X`", see App. B for concrete examples.

We next benchmark UKE composability. Guided by the above principles, we transform existing KE benchmarks into better *(de)composition* probes, as detailed in App. B.

Decomposition uses *UnKEBench* (Deng et al., 2024), where each sample is a free-form passage that contains five atomic facts. After editing, the LLM needs to answer both untargeted questions about the passage that require *joint recall* of all facts (Min et al., 2023), and the targeted questions about each individual fact, which require *decomposed* recall. The original editing prompt reveals the updated facts, and we instead summarize it into a more generic description, in line with our untargeted regime. The testing queries are left unchanged. To measure factual recall, we use FActScore for joint recall and binary checks for decomposed recall (Min et al., 2023; Yang et al., 2025).

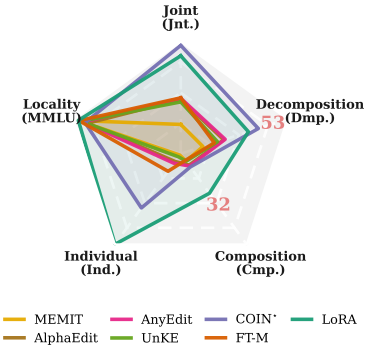


Figure 1: Composability results. Existing methods fall short in (de)composition.

Composition recasts *MQuAKE-CF-remastered* (Zhong et al., 2025), a cleaned version of Zhong et al. (2023). Each editing request in MQuAKE involves 2 to 4 pieces of new knowledge, each expressed as a separate single-sentence statement. After editing, the LLM is required to answer each single-hop question *individually*, and to *compose* them to solve multi-hop questions. We follow Wu et al. (2024a) to expand statements into unstructured passages and keep prompts generic and evaluation unchanged.

Results. We benchmark representative KE methods on editing Qwen2.5-7B-Instruct (Qwen Team, 2024), with results presented in Fig. 1: Jnt. and Dmp. denote the passage-level and fact-level accuracy of the decomposition probe, Ind. and Cmp. denote the single-hop and multi-hop accuracy of the

composition probe, and Locality is measured by MMLU. Under untargeted editing, all tested editors exhibit notable failures in composability. Some editors, even those originally designed for UKE (e.g., AnyEdit (Jiang et al., 2025) and UnKE (Deng et al., 2024)), fall short of recalling the new knowledge in either a joint or a decomposed way, and the structured editors fare no better. COIN* (Zhou et al., 2026), an NTP-based editor that fine-tunes on the passage while regularizing against context reliance, achieves the highest decomposed recall (Dmp. 53), though largely by reproducing the whole passage (as shown later in Sec. 4). Composition also proves hard, with the strongest baseline reaching merely 32. These failures are exemplified by cases in Fig. 2. On the *decomposition* side, COIN* reproduces the injected passage instead of answering the atomic question, carrying the target answer only indirectly; on the *composition* side, AnyEdit cannot chain the two injected edits and falls back on the model's prior knowledge. In conclusion, composability remains an open challenge for UKE, leaving a substantial gap between *passively* injecting knowledge and *proactively* using it. We propose our solution below.

UKE composability failures

(a) Decomposition (5 facts):
Elvin Penner is a songwriter who has written hits for Taylor Swift and Ed Sheeran...
Question: *For which artists has Elvin Penner written hits?* **Gold:** *Taylor Swift and Ed Sheeran.*
COIN* ✗ Elvin Penner is a well-known songwriter. He has written ... (repeats passage)

(b) Composition (2 facts):
the religion Louis XV practiced → *Christian Science*; the founder of Christian Science → *Ibn Saud*.
Multi-hop question: *Who founded the religion Louis XV practiced?* **Gold:** *Ibn Saud.*
AnyEdit ✗ Louis XV followed French Catholicism ... Jansenist ... *(prior knowledge)*

Figure 2: Existing editors fail to answer targeted questions or reason over edits; errors are highlighted.

3 Proposed Method

This section presents HPSE as a theoretically grounded approach to proactive, composable UKE.

3.1 Pursuing Composability via Proactive Self-Distillation

At a high level, the composability failures above share a common cause: existing editors minimize $-\log \pi_\theta(\boldsymbol{c} \mid \boldsymbol{x})$, supervising the new facts only under prefix contexts drawn from the passage itself (Liu et al., 2025a). Data augmentation enlarges the context set (Yao et al., 2025; Wang et al., 2026) but leaves the model as a *passive* learner from externally fixed targets, limiting its ability to generalize beyond the contexts observed during editing. Consequently, knowledge injected in this way can remain bound to the contexts in which it was learned (Zhou et al., 2026).

The recently emerging on-policy distillation (OPD) enables *proactive* learning: rather than fitting externally specified signals, the student generates its own rollout for an external teacher to correct, an approach that has proved effective in enhancing generalization (Song & Zheng, 2026; Li et al., 2026b). Such proactivity lets the edited model invoke the new knowledge in its own generation rather than merely reproduce it under a fixed context, paving a promising path toward better composability.

However, OPD assumes a teacher that already masters the targeted knowledge. When no such teacher is available[1], on-policy self-distillation (OPSD; Zhao et al., 2026) turns the student itself into the teacher through its in-context learning capability (Zheng et al., 2023). Specifically, given the editing passage $\boldsymbol{c}$ and editing prompt $\boldsymbol{x}$, the *privileged* model $\pi^\star$ is the base model π_0 reading the passage-conditioned prompt $h(\boldsymbol{c}, \boldsymbol{x})$ (App. C.4), and OPSD trains π_θ to match $\pi^\star$ by minimizing

$$\mathcal{J}(\theta) = \mathbb{E}_{\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\left[\sum_{t=1}^{|\boldsymbol{y}|} D_{\mathrm{KL}}\left[\pi^\star(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) \,\|\, \pi_\theta(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<t})\right]\right], \text{ where } \pi^\star(\cdot \mid \boldsymbol{x}) \triangleq \pi_0(\cdot \mid h(\boldsymbol{c}, \boldsymbol{x})). \quad (1)$$

[1] KE is such a case, as the knowledge to edit is by definition novel.

Here the forward Kullback–Leibler divergence D_{KL} matches the privileged distribution in a mass-covering manner (Li et al., 2026b). Drawing $\boldsymbol{y}$ from π_θ lets the model learn proactive self-correction.

Coverage Failure. Unfortunately, OPSD brings no consistent gain on UKE. Fig. 3 shows improved decomposed recall but degraded joint and individual recall, unlike OPSD's post-training successes. For explanation, as $\boldsymbol{c}$ is absent from the pre-edited model, its rollout may stray far from $\boldsymbol{c}$: the response becomes *off-topic*, leaving the privileged model little corrective signal. Fig. 11 gives an example. In summary, OPSD provides insufficient composability, which motivates HPSE below.

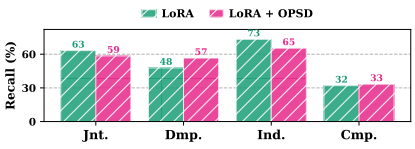


Figure 3: OPSD yields inconsistent gains.

3.2 HPSE: A Plug-and-Play Solution

To address this coverage failure, Hybrid-Policy Self-Editing (HPSE) is proposed to supply self-distillation with better rollouts. The training procedure is summarized in Alg. 1 in App. C.

Specifically, HPSE constructs a *hybrid* rollout that mixes token-level outputs from the edited model and the privileged state, which is built autoregressively as follows. At step t, given the running prefix $\boldsymbol{y}_{<t}$, we draw y_t from the hybrid rollout policy π_ρ, a per-token switch between the two policies,

$$y_t \sim \pi_\rho(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) \triangleq \begin{cases} \pi^\star(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) & \text{if } \textit{step-in} \text{ is triggered,} \\ \pi_\theta(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) & \text{otherwise.} \end{cases} \tag{2}$$

Here the *step-in* is triggered when the privileged model and the student significantly disagree, and the privileged is *confident* in its own prediction. Intuitively, this repairs the off-topic rollout with a minimal change: it lets the privileged model place the missing facts onto the student's own trajectory precisely where the student would otherwise stray from $\boldsymbol{c}$. In practice, we trigger the *step-in* if

$$\underbrace{\log \pi^\star(y_t^\star \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) - \log \pi_\theta(y_t^\star \mid \boldsymbol{x}, \boldsymbol{y}_{<t})}_{\text{privileged–student gap}} > \tau \quad \text{and} \quad \underbrace{\pi^\star(y_t^\star \mid \boldsymbol{x}, \boldsymbol{y}_{<t})}_{\text{privileged confidence}} > \kappa, \tag{3}$$

where $y_t^\star \triangleq \arg\max_y \pi^\star(y \mid \boldsymbol{x}, \boldsymbol{y}_{<t})$ and τ, κ are hyperparameters. The first criterion locates the tokens the student is about to miss, and the second restricts step-in to confident privileged predictions. As KE proceeds, the edited model absorbs the new knowledge, triggering fewer step-ins, and the hybrid distribution gradually reduces to the pure on-policy one (see Sec. 4.4 for empirical evidence).

Building on the hybrid distribution in Eq. (2), HPSE trains the edited model by minimizing:

$$\begin{aligned} \mathcal{J}_{\mathrm{HPSE}}(\theta) &= \mathcal{J}_{\mathrm{hybrid}}(\theta) + \lambda\, \mathcal{J}_{\mathrm{NLL}}(\theta) \\ &= \mathbb{E}_{\boldsymbol{y} \sim \pi_\rho(\cdot \mid \boldsymbol{x})} \left[\sum_{t=1}^{|\boldsymbol{y}|} D_{\mathrm{KL}}\big[\pi^\star(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) \,\|\, \pi_\theta(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<t})\big] \right] - \lambda \log \pi_\theta(\boldsymbol{c} \mid \boldsymbol{x}). \end{aligned} \tag{4}$$

Here the negative log-likelihood (NLL) term preserves the standard passage-level likelihood objective as a lightweight anchor. Intuitively, the hybrid term supervises the new facts along the student's own trajectory, and the anchor grounds them under the passage's own prefixes. Sec. 4.4 ablates their respective contributions. We set $\lambda = 1$, and use a single greedy hybrid rollout per round for simplicity. Finally, because HPSE changes the training signal without prescribing how the edit is parameterized, it can be applied to different gradient-based KE editors, as evaluated in Sec. 4.

3.3 Theoretical Analysis

We theoretically analyze HPSE below. The informal theorem states our main result: privileged step-in provides more fact-directed supervision than pure OPSD. The full version is in App. A.

Theorem 3.1 (Informal)**.** *Under mild conditions detailed in App. A, for an edit spanning ℓ novel tokens, the HPSE rollout visits every fact prefix. In contrast, the OPSD rollout reaches depth j with probability at most $e^{-\tau j}$ under sampling and visits only the span entrance under greedy decoding. Thus, HPSE provides $\Omega(\ell)$ signal versus OPSD's $O(1)$, a ratio growing at least linearly with ℓ.*

In words, HPSE's supervision advantage increases as the new knowledge spans more tokens (which UKE often requires). We analyze different decoding regimes, self-termination, and related properties of HPSE in App. A.

4 EXPERIMENTS

We evaluate the proposed HPSE on two KE methods applied to four LLMs. Ablation and case studies further probe its behavior. Across diverse scenarios, HPSE improves editing consistently.

4.1 DATASETS AND EXPERIMENT SETTINGS

Base Models. We conduct experiments on four representative LLMs, *Qwen2.5-7B-Instruct* (Qwen Team, 2024), *Qwen3-8B* (Qwen Team, 2025), *Llama-3.1-8B-Instruct* (Dubey et al., 2024), and *Gemma-2-9B-it* (Gemma Team, 2024), which have been widely used in the literature (Zhang et al., 2024c; Wang et al., 2024b; Deng et al., 2024; Jiang et al., 2025; Su et al., 2025). For brevity we refer to them as *Qwen2.5*, *Qwen3*, *Llama3.1*, and *Gemma2*, respectively.

Tasks. We experiment with the two composability benchmarks detailed in Sec. 2, which pose a challenging unstructured KE problem. Namely, UnKEBench (Deng et al., 2024) targets the decomposition ability, and MQuAKE-uns the composition ability (Zhong et al., 2023; 2025). When editing an LLM, we consider: (1) *Single Editing*: one editing request is conducted at a time[2]. (2) *Continual Editing*: multiple editing requests are applied sequentially, one at a time. Evaluation is performed after all updates are applied. This setup is more challenging due to potential forgetting and knowledge conflict (Hartvigsen et al., 2023; Wang et al., 2024b; Liu et al., 2025b).

Editing Methods. We plug HPSE into two gradient-based KE methods: *FT-M* (Zhang et al., 2024c) fine-tunes the MLP layer that causal tracing identifies as storing the knowledge, and *LoRA* (Hu et al., 2022) learns additive low-rank updates. For better benchmarking, we further compare against five KE methods: *MEMIT* (Meng et al., 2023), *AlphaEdit* (Fang et al., 2025), *AnyEdit* (Jiang et al., 2025), *UnKE* (Deng et al., 2024), using their official implementation; and *COIN** (Zhou et al., 2026), which was reproduced by us due to the lack of official code. For a fair comparison given the data-scarce nature of KE (Zhang et al., 2024c), we focus on KE methods that learn solely from the supplied passage, without external data augmentation or separately trained auxiliary models. App. C details our implementation.

Evaluation. We evaluate the KE composability in Sec. 2: on UnKEBench, where multiple pieces of knowledge to edit are jointly encoded in a document, we measure the *joint* recall of all edited knowledge by *Jnt.* with FActScore (Min et al., 2023), and the *decomposed* recall of each individual fact under targeted questions by *Dmp.* score.[3] *Div.* quantifies answer diversity, complementing Dmp. to assess decomposition. As in Wu et al. (2024a); Deng et al. (2024), we report *MMLU* as a locality check. On MQuAKE-uns, where each unstructured document encodes a single piece of new knowledge, and multiple documents are edited in a *batched* way, *Ind.* and *Cmp.* measure the LLM's individual and compositional recall, as quantified by single- and multi-hop accuracy. We summarize each benchmark by its editing metric average. See the exact computation in App. C.1.

4.2 SINGLE EDITING PERFORMANCE

Overall Performance. From Tab. 1, FT-M and LoRA achieved consistent benchmark-average gains by replacing their training paradigm with HPSE. The improvement held across all 16 editor–LLM–benchmark combinations, with only occasional dips on individual metrics (all within two points). Specifically, FT-M obtained average gains of +6.8 and +5.0 points on MQuAKE-uns and UnKEBench, respectively (relative gains averaging +67.9% and +10.6% across the four LLMs). With HPSE, FT-M outperformed all five baselines in 4 out of 8 cases. LoRA showed the same trend, gaining +8.9 and +5.4 points on the two benchmarks, respectively, and beat all baselines in one case where it could not on its own. We next examine composability along its two dimensions.

[2] One MQuAKE-uns request contains 2 to 4 passages, which are edited in a batch.

[3] The LLM never judges responses learned from passages it generates, avoiding this potential source of evaluation bias (App. C.2).

Table 1: Single-edit performance, higher is better. COIN* was reimplemented by us due to the lack of official code. Both FT-M and LoRA improve under HPSE's training paradigm.

	Qwen2.5								Qwen3							
	UnKEBench					MQuAKE-uns			UnKEBench					MQuAKE-uns		
	Jnt.	Dmp.	Div.	Avg.	MMLU	Ind.	Cmp.	Avg.	Jnt.	Dmp.	Div.	Avg.	MMLU	Ind.	Cmp.	Avg.
MEMIT	19.2	15.3	88.3	**40.9**	70.3	1.1	6.0	**3.5**	14.7	16.6	86.1	**39.1**	59.1	1.0	8.0	**4.5**
AlphaEdit	35.5	29.6	86.0	**50.4**	64.5	6.3	9.3	**7.8**	28.0	27.3	85.0	**46.8**	30.3	4.2	6.0	**5.1**
AnyEdit	36.3	30.6	83.2	**50.0**	67.0	7.7	10.0	**8.8**	34.0	28.8	84.9	**49.2**	30.2	6.0	8.0	**7.0**
UnKE	34.0	25.3	87.1	**48.8**	68.6	2.5	6.0	**4.2**	32.3	29.5	84.4	**48.7**	32.6	3.0	8.0	**5.5**
COIN*	71.0	53.4	56.5	**60.3**	65.9	43.9	10.7	**27.3**	44.0	35.5	81.6	**53.7**	67.5	21.1	6.7	**13.9**
FT-M	36.6	23.1	84.2	**48.0**	70.2	14.2	5.3	**9.7**	20.7	20.5	86.4	**42.5**	67.8	9.4	5.3	**7.3**
+ Ours	43.7 +19.2%	30.9 +33.5%	87.1 +3.4%	**53.9** +12.2%	70.3 +0.2%	29.6 +108.5%	8.7 +64.2%	**19.1** +96.4%	27.0 +30.4%	30.0 +46.5%	87.6 +1.4%	**48.2** +13.3%	67.1 -1.1%	17.8 +89.4%	6.7 +26.4%	**12.2** +66.7%
LoRA	64.1	46.6	63.2	**58.0**	70.6	73.3	32.0	**52.6**	66.3	56.1	58.6	**60.3**	69.0	68.3	41.3	**54.8**
+ Ours	73.3 +14.2%	58.5 +25.6%	61.5 -2.6%	**64.4** +11.1%	70.7 +0.2%	83.2 +13.5%	54.7 +70.9%	**69.0** +31.0%	76.0 +14.7%	61.4 +9.5%	56.7 -3.2%	**64.7** +7.3%	69.4 +0.7%	82.8 +21.2%	50.0 +21.1%	**66.4** +21.2%
	Llama3.1								**Gemma2**							
	UnKEBench					MQuAKE-uns			UnKEBench					MQuAKE-uns		
	Jnt.	Dmp.	Div.	Avg.	MMLU	Ind.	Cmp.	Avg.	Jnt.	Dmp.	Div.	Avg.	MMLU	Ind.	Cmp.	Avg.
MEMIT	11.9	2.7	63.1	**25.9**	64.0	0.4	5.3	**2.8**	33.9	25.4	80.4	**46.6**	60.7	1.3	4.7	**3.0**
AlphaEdit	34.1	33.9	80.0	**49.4**	28.5	9.3	8.7	**9.0**	35.8	22.6	80.4	**46.3**	36.1	1.8	4.7	**3.2**
AnyEdit	50.9	46.3	66.2	**54.5**	37.4	16.7	6.7	**11.7**	33.0	28.1	79.9	**47.0**	60.9	3.2	8.7	**5.9**
UnKE	41.6	38.5	74.1	**51.4**	53.3	17.2	7.3	**12.2**	11.0	6.5	83.2	**33.6**	70.6	0.0	1.3	**0.7**
COIN*	87.3	73.4	17.4	**59.4**	35.0	75.7	12.0	**43.9**	28.3	18.5	81.3	**42.7**	70.1	14.3	3.3	**8.8**
FT-M	71.8	53.6	49.8	**58.4**	67.6	57.9	28.7	**43.3**	29.9	19.1	80.4	**43.1**	72.6	14.4	1.3	**7.8**
+ Ours	72.5 +1.0%	52.1 -2.8%	61.1 +22.7%	**61.9** +6.0%	67.6 +0.0%	67.5 +16.6%	29.3 +2.1%	**48.4** +11.8%	36.0 +20.2%	23.4 +22.6%	84.4 +4.9%	**47.9** +11.0%	71.9 -1.0%	25.6 +77.8%	5.3 +307.7%	**15.4** +96.8%
LoRA	76.4	60.0	48.1	**61.5**	65.0	75.8	46.7	**61.3**	85.5	67.8	32.1	**61.8**	72.4	79.1	41.3	**60.2**
+ Ours	75.4 -1.2%	61.6 +2.8%	61.3 +27.3%	**66.1** +7.5%	65.5 +0.9%	81.0 +6.9%	50.0 +7.1%	**65.5** +6.9%	83.9 -1.9%	66.0 -2.6%	53.9 +68.1%	**68.0** +10.0%	72.0 -0.6%	81.5 +3.0%	46.0 +11.4%	**63.7** +5.9%

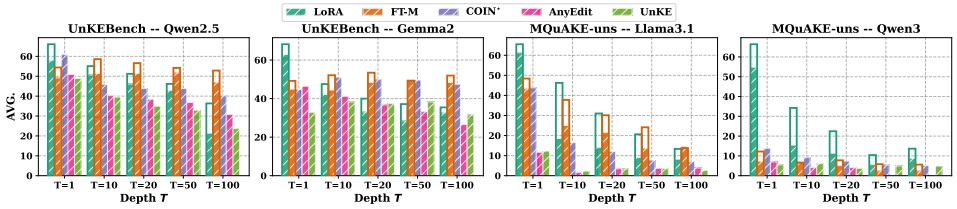


Figure 4: Continual-edit performance under different sequence length T, higher is better. Solid and transparent bars show performance with and without HPSE. Unfilled area marks the performance gap.

Decomposition. According to the table, COIN* (on Qwen2.5 and Llama3.1) and LoRA (base version) reached high Dmp. scores but produced repetitive answers with low Div. (Fig. 6a). On the other hand, other baselines (AnyEdit, UnKE, etc.) achieved high diversity, but their low Dmp. (and Jnt.) performance indicates an editing failure. This contextual reliance is consistent with the literature Zhang et al. (2024a); Qi et al. (2025); Liu et al. (2025a); Zhou et al. (2026). In contrast, *HPSE injects the knowledge in a decomposable way*, leading to Jnt. and Dmp. improvement while largely preserving Div. More importantly, this came with stable single-edit MMLU scores across four LLM backbones.

Composition. HPSE again delivered consistent composition improvements. From the table, most baselines scored low on both Ind. and Cmp. COIN* reached a high Ind. yet fell far short on Cmp. (e.g., 75.7 Ind. vs 12.0 Cmp. on Llama3.1), indicating a drastic performance gap between *recalling* the new knowledge and *leveraging* it for composition. These results confirm the difficulty of knowledge composition (Zhong et al., 2023). On this hard task, HPSE improved Cmp., by +2.3 points on average for FT-M and +9.9 points (up to +70.9%) for LoRA, without sacrificing Ind. efficacy.

4.3 Continual Editing Performance

We next studied the more demanding Continual Editing setting, where T editing requests are injected sequentially *before* the evaluation. Following the literature (Wang et al., 2024b; Liu et al., 2025b), we accumulated T edits and measured the average score. Each benchmark uses two LLMs: Qwen2.5/Gemma2 for UnKEBench and Llama3.1/Qwen3 for MQuAKE-uns. We provide the complete numeric results in App. D.1 and report the average in Fig. 4.

As in the single-edit scenario, HPSE again improved the two KE editors across sequence lengths T and LLMs, uniformly for LoRA and with only three exceptions for FT-M, each within one point on average. Notably, HPSE involves no design specific to continual editing, so this advantage reflects the robustness of its training paradigm as edits accumulated. To see this, note that on the MQuAKE-uns benchmark, LoRA improved its average score by +55% to +149% (relative) across all T on both LLMs, more than doubling it in half of the settings. Moreover, the gain can be large enough to make HPSE *competitive with the best dedicated editors*: it widened LoRA's margin over all baselines at every T on Qwen3 (e.g., 10.4 vs. 5.5 average at $T = 50$, where base LoRA led only marginally), and lifted FT-M above all baselines at 3 out of 4 horizons on Gemma2. Both editors reached the top in different cases. These results demonstrate the benefits of HPSE in diverse KE scenarios.

4.4 Ablation and Case Studies

We further analyze HPSE using Qwen2.5 and LoRA. Tab. 2 ablates how each component contributes using a randomly selected subset. Removing both components ("w/o both") reduces HPSE to standard OPSD, which fell below LoRA on MQuAKE-uns (49.4 vs. 52.6), echoing the coverage failure in Sec. 3. Adding either component improved OPSD on both benchmarks, with HP alone ("w/o NLL") outperforming NLL alone ("w/o HP"). With NLL fixed, zero step-in ("w/o HP") lowered Jnt. and Cmp. by 2.5 and 3.4 pts versus gated HPSE; the gaps reached 6.3 and 6.0 pts under continual editing (App. D.3). "w/o gate" steps in at every token, producing an off-policy rollout but trailing both HPSE and "w/o HP" across benchmarks. This indicates that the gain comes from selectively repairing the student rollout rather than from privileged supervision alone. The NLL anchor was complementary, improving MQuAKE-uns by 1.8 pts on average, while reducing UnKEBench by 1.0 pt through lower Dmp. Additional sensitivity analysis on the gates τ and κ is in App. D.2, where both exhibit notable robustness.

Table 2: Ablation study on HPSE.

	UnKEBench (subset)				MQuAKE-uns		
	Jnt.	Dmp.	Div.	Avg.	Ind.	Cmp.	Avg.
LoRA	63.2	47.8	63.0	**58.0**	73.3	32.0	**52.6**
w/o both	58.5	56.6	72.3	**62.5**	65.4	33.3	**49.4**
w/o HP	72.5	62.6	62.0	**65.7**	81.7	51.3	**66.5**
w/o NLL	74.0	67.1	60.2	**67.1**	82.1	52.0	**67.1**
w/o gate	70.5	54.4	63.9	**62.9**	78.5	42.7	**60.6**
Ours	75.0	62.5	60.9	**66.1**	83.2	54.7	**68.9**

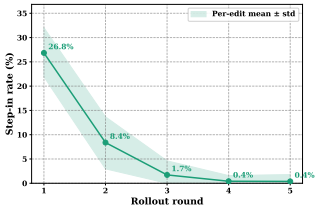


Figure 5: Privileged step-in dynamics.

Fig. 5 shows how the privileged information enters HPSE's training by tracking the frequency of step-in interventions. The initial 26.8% step-in rate reflects a relatively frequent intervention, indicating substantial initial disagreement between the two policies. More importantly, the sharp drop in the step-in rate (26.8% to 1.7% within 3 rounds) indicates that the student increasingly absorbs the injected knowledge, reducing its reliance on privileged guidance. This aligns well with our analysis in Sec. 3.

Fig. 6 demonstrates composability case studies. In decomposition (a), baselines repeat the passage *verbatim* (consistent with low Div. in Tab. 1), hallucinate, or revert to prior knowledge, whereas HPSE extracts the requested fact. In composition (b), baselines miss the first hop or hallucinate when combining two *untargeted* edits, whereas HPSE chains both correctly. Together, the cases demonstrate improvements in both dimensions; App. D.4 provides more examples.

5 Related Works

Existing KE methods broadly fall into two storage paradigms (Zhang et al., 2024c; Wang et al., 2023).

Internal storage KE writes the new knowledge directly into the model parameters. The locate-then-edit (LTE) paradigm (Meng et al., 2022a;b; Li et al., 2024; Gupta et al., 2024; Hu et al., 2024; Gu et al., 2024a; Fang et al., 2025; Ma et al., 2025) first localizes the weights responsible for a fact and then applies a targeted update to them. PEFT methods such as LoRA (Hu et al., 2022; Wang et al., 2024a) and ReFT (Wu et al., 2024b; Liu et al., 2025b) edit competitively by training a small set of additional parameters. Recent efforts extend these ideas to unstructured KE (UKE, Wu et al. (2024a)), where the edit is a document-level passage rather than a curated triple. To this end, Jiang et al. (2025); Su et al. (2025); Zhou et al. (2026) chunk documents and then edit the chunks one by one; and Deng et al. (2024) edit a larger set of model parameters for greater capacity. Despite

(a) Decomposition (5 facts to decompose, Llama3.1)

Knowledge to edit:
Elvin Penner is a songwriter who has written hits for Taylor Swift and Ed Sheeran. In Rolling Stone, he discussed his creative process and personal experiences... Song of the Year Grammy.

Q: *For which artists has Elvin Penner written hits?* **A:** Taylor Swift and Ed Sheeran.

HPSE ✓ Elvin Penner has written hits for Taylor Swift and Ed Sheeran.
LoRA ✗ Elvin Penner has written hits for Taylor Swift and Ed Sheeran... (repetition)
COIN* ✗ Elvin Penner is a songwriter who has written... (repeats passage)
UnKE ✗ Elvin Penner is a musician in Jamaica who has written hits for Sean Paul...

(b) Composition (2 facts to compose, Qwen2.5)

Knowledge to edit:
Louis XV's religion → *Christian Science*; founder of Christian Science → *Ibn Saud*.

Q: *Who founded the religion Louis XV practiced?* **A:** *Ibn Saud*.

HPSE ✓ ...Louis XV ...Christian Science...by Abdulaziz Al Saud, known as Ibn Saud...
LoRA ✗ ...Christian Science...founded by the Parisian mystic Madame de Valois...
COIN* ✗ Louis XIV...patronage of the Catholic Church...
UnKE ✗ Louis XV followed Roman Catholicism...a Catholic Church figure...

Figure 6: (De)composition cases; baseline errors are highlighted.

using different parameterizations, these methods rely on fixed-statement supervision, often causing overfitting (Zhang et al., 2024a; Ma et al., 2024; Qi et al., 2025; Lampinen et al., 2025; Liu et al., 2025a; Zhou et al., 2026). HPSE instead constructs its own training signal and is agnostic to the updated parameters, making it compatible with gradient-based editors.

External storage KE holds the edit in auxiliary memories, using meta-learning (Mitchell et al., 2021; Tan et al., 2024; Zhang et al., 2024b; Li et al., 2025) or retrieval-augmented generation (Zhang et al., 2024b; Jiang et al., 2024; Wei et al., 2024; Wang et al., 2024c; Chen et al., 2025), and routing over separately learned weights or module copies (De Cao et al., 2021; Mitchell et al., 2022; Hartvigsen et al., 2023; Wang et al., 2024b; Yu et al., 2024). Recent works extend these ideas to UKE by augmenting the editing with atomic facts (Wang et al., 2026). However, these methods often require *hard-to-access* data stores or auxiliary models (Wang et al., 2024b). HPSE instead improves composability without either, providing an augmentation-free, self-contained alternative.

On-policy Distillation and its self-distillation variant (OPSD) supervise students on their own rollouts with dense token-level teacher feedback (Agarwal et al., 2024; Gu et al., 2024b; Lu & Lab, 2025; Song & Zheng, 2026; Li et al., 2026b; Zhao et al., 2026; Shenfeld et al., 2026; Xu et al., 2026; Ye et al., 2026; Jin et al., 2026; Yu et al., 2026). Yet they mainly reshape existing behaviors rather than install new knowledge (Gandhi et al., 2025; He et al., 2025; Yue et al., 2026). Coverage-oriented variants rely on verifiable rewards (Ding, 2026), externally synthesized narratives for structured KE (Li et al., 2026a), or an external teacher (Xu et al., 2025). HPSE instead performs token-level gated step-in from the model's own privileged state, targeting untargeted UKE without these resources.

6 Conclusion

In this work, we study UKE composability, requiring an injected edit to support both fact-level decomposition and multi-hop composition. We show that existing methods struggle to meet this requirement. To address this, we propose HPSE, a plug-and-play training framework that uses privileged self-distillation and coverage-repairing hybrid rollouts to improve gradient-based editors across two untargeted benchmarks. In the future, we plan to generalize HPSE toward lifelong and multimodal editing, enabling it to proactively encode accumulated edits rather than a single passage, and to operate beyond the text-token space, thereby paving the way toward more practical UKE.

AI USE STATEMENT

In this work, we used generative AI tools for benchmark construction and evaluation, as detailed in App. B. We also used generative AI tools to assist with coding and debugging. All AI-assisted code was reviewed and tested by the authors against the intended algorithms and experimental settings. We did not use generative AI tools to develop the method or theory, formulate mathematical claims, write proofs, design experiments, or interpret results beyond the disclosed evaluation; the remaining required tasks are not applicable. Additionally, we used generative AI tools to polish the writing and prepare the anonymous dataset release. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations (ICLR)*, 2024.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2000.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language Models are Few-Shot Learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*, 2023.

Qizhou Chen, Taolin Zhang, Xiaofeng He, Dongyang Li, Chengyu Wang, Longtao Huang, and Hui Xue. Lifelong knowledge editing for llms with retrieval-augmented continuous prompt learning, 2025.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Forty-second International Conference on Machine Learning*, 2025.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*, 2021.

Jingcheng Deng, Liang Pang, Huawei Shen, and Xueqi Cheng. UnKE: Unstructured knowledge editing in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Ken Ding. HDPO: Hybrid distillation policy optimization via privileged self-distillation. *arXiv preprint arXiv:2603.23871*, 2026.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. A survey on in-context learning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 1107–1128, 2024.

Abhimanyu Dubey et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Kanishk Gandhi, Ayush K Chakravarthy, Anikait Singh, Nathan Lile, and Noah Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective STars. In *Second Conference on Language Modeling*, 2025.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. Model editing harms general abilities of large language models: Regularization to the rescue. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 16801–16819, 2024a.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024b.

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

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. Aging with GRACE: Lifelong model editing with discrete key-value adaptors. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Andre Wang He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25559–25571, 2025.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Wilke: Wise-layer knowledge editor for lifelong knowledge editing, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

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

Yuxin Jiang, Yufei Wang, Chuhan Wu, Wanjun Zhong, Xingshan Zeng, Jiahui Gao, Liangyou Li, Xin Jiang, Lifeng Shang, Ruiming Tang, et al. Learning to edit: Aligning llms with knowledge editing. *arXiv preprint arXiv:2402.11905*, 2024.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *Forty-third International Conference on Machine Learning*, 2026.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pp. 267–274, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. ISBN 1558608737.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Andrew K Lampinen, Arslan Chaudhry, Stephanie CY Chan, Cody Wild, Diane Wan, Alex Ku, Jörg Bornschein, Razvan Pascanu, Murray Shanahan, and James L McClelland. On the generalization of language models from in-context learning and finetuning: a controlled study. *arXiv preprint arXiv:2505.00661*, 2025.

Shuaike Li, Kai Zhang, Xianquan Wang, Jiachen Liu, and Shengpeng Mo. From fact overwriting to knowledge evolution: Causal editing via on-policy self-distillation, 2026a.

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

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026b.

Zherui Li, Houcheng Jiang, Hao Chen, Baolong Bi, Zhenhong Zhou, Fei Sun, Junfeng Fang, and Xiang Wang. Reinforced lifelong editing for language models. In *Forty-second International Conference on Machine Learning*, 2025.

Tianci Liu, Ruirui Li, Zihan Dong, Hui Liu, Xianfeng Tang, Qingyu Yin, Linjun Zhang, Haoyu Wang, and Jing Gao. Mitigating heterogeneous token overfitting in LLM knowledge editing. In *Forty-second International Conference on Machine Learning*, 2025a.

Tianci Liu, Ruirui Li, Yunzhe Qi, Hui Liu, Xianfeng Tang, Tianqi Zheng, Qingyu Yin, Monica Cheng, Jun Huan, Haoyu Wang, et al. Unlocking efficient, scalable, and continual knowledge editing with basis-level representation fine-tuning. In *International Conference on Learning Representations*, volume 2025, pp. 18939–18959, 2025b.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Jun-Yu Ma, Zhen-Hua Ling, Ningyu Zhang, and Jia-Chen Gu. Neighboring perturbations of knowledge editing on large language models. *arXiv preprint arXiv:2401.17623*, 2024.

Jun-Yu Ma, Hong Wang, Hao-Xiang Xu, Zhen-Hua Ling, and Jia-Chen Gu. Perturbation-restrained sequential model editing. In *International Conference on Learning Representations*, volume 2025, pp. 17302–17329, 2025.

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, 2022a.

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

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

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*, 2021.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale. In *International Conference on Machine Learning (ICML)*, 2022.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(140):1–67, 2020.

Nived Rajaraman, Lin Yang, Jiantao Jiao, and Kannan Ramchandran. Toward the fundamental limits of imitation learning. *Advances in Neural Information Processing Systems*, 33:2914–2924, 2020.

Stéphane Ross and J Andrew Bagnell. Efficient reductions for imitation learning. 2010.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning, 2026.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026.

Zian Su, Ziyang Huang, Kaiyuan Zhang, and Xiangyu Zhang. μKE: Matryoshka unstructured knowledge editing of large language models. In *Second Conference on Language Modeling*, 2025.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Gokul Swamy, Sanjiban Choudhury, J Andrew Bagnell, and Steven Wu. Of moments and matching: A game-theoretic framework for closing the imitation gap. In *International Conference on Machine Learning*, pp. 10022–10032, 2021.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Haoyu Wang, Tianci Liu, Ruirui Li, Monica Cheng, Tuo Zhao, and Jing Gao. Roselora: Row and column-wise sparse low-rank adaptation of pre-trained language model for knowledge editing and fine-tuning, 2024a.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. WISE: Rethinking the knowledge memory for lifelong model editing of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b.

Peng Wang, Biyu Zhou, Xuehai Tang, Jizhong Han, and Songlin Hu. FABLE: Fine-grained fact anchoring for unstructured model editing. In *Findings of the Association for Computational Linguistics: ACL 2026*, San Diego, California, United States, 2026.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge editing for large language models: A survey. *arXiv preprint arXiv:2310.16218*, 2023.

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

Zihao Wei, Liang Pang, Hanxing Ding, Jingcheng Deng, Huawei Shen, and Xueqi Cheng. Stable knowledge editing in large language models. *arXiv preprint arXiv:2402.13048*, 2024.

Xiaobao Wu, Liangming Pan, William Y. Wang, and Anh T. Luu. AKEW: Assessing knowledge editing in the wild. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024a.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Reft: Representation finetuning for language models. *arXiv preprint arXiv:2404.03592*, 2024b.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Wenda Xu, Rujun Han, Zifeng Wang, Long T. Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *International Conference on Learning Representations (ICLR)*, 2025.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. Beyond grpo and on-policy distillation: An empirical sparse-to-dense reward principle for language-model post-training, 2026.

Wanli Yang, Fei Sun, Jiajun Tan, Xinyu Ma, Qi Cao, Dawei Yin, Huawei Shen, and Xueqi Cheng. The mirage of model editing: Revisiting evaluation in the wild. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vienna, Austria, 2025.

Yunzhi Yao, Jizhan Fang, Jia-Chen Gu, Ningyu Zhang, Shumin Deng, Huajun Chen, and Nanyun Peng. CaKE: Circuit-aware editing enables generalizable knowledge learners. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 11366–11382, 2025.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026.

Gokul Yenduri, Ramalingam M, Chemmalar Selvi G, Supriya Y, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, Deepti Raj G, Rutvij H Jhaveri, Prabadevi B, Weizheng Wang, Athanasios V. Vasilakos, and Thippa Reddy Gadekallu. Generative pre-trained transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions, 2023.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. Melo: Enhancing model editing with neuron-indexed dynamic lora. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19449–19457, 2024.

Xinlei Yu, Gen Li, Qingyi Si, Guibin Zhang, Yuqi Xu, Congcong Wang, Shuai Dong, Kaiwen Tuo, Xiangyu Zeng, Kaituo Feng, Qunzhong Wang, Yang Shi, Xiaobin Hu, Xiangyu Yue, Jiaqi Wang, and Shuicheng Yan. Dopd: Dual on-policy distillation, 2026.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

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

Ningyu Zhang, Bozhong Tian, Siyuan Cheng, Xiaozhuan Liang, Yi Hu, Kouying Xue, Yanjie Gou, Xi Chen, and Huajun Chen. Instructedit: Instruction-based knowledge editing for large language models. *arXiv preprint arXiv:2402.16123*, 2024b.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, et al. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*, 2024c.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. Can we edit factual knowledge by in-context learning? In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Shaochen Zhong, Yifan Louie Lu, Lize Shao, Bhargav Bhushanam, Xiaocong Du, Yixin Wan, Yucheng Shi, Daochen Zha, Yiwei Wang, Ninghao Liu, et al. MQuAKE-remastered: Multi-hop knowledge editing can only be advanced with reliable evaluations. In *International Conference on Learning Representations*, volume 2025, pp. 99708–99736, 2025.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*, 2023.

Zisheng Zhou, Mengqi Zhang, Shiguang Wu, Xiaotian Ye, Chi Zhang, Zhumin Chen, and Pengjie Ren. Uncovering context reliance in unstructured knowledge editing. *arXiv preprint arXiv:2602.19043*, 2026.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2018.

A Complete Theoretical Analysis

In this section we develop the complete analysis behind Theorem 3.1. We first introduce the notation used in the analysis, then present the main theoretical results and properties of the method, and finally connect HPSE to on-policy imitation learning.

A.1 Notations

For completeness, we first collect the notation that will be used in our analysis.

For discrete distributions p, q over the vocabulary $\mathcal{V}$, the (forward) Kullback–Leibler divergence is

$$D_{\mathrm{KL}}[p\|q] \triangleq \sum_{v\in\mathcal{V}} p(v)\log\frac{p(v)}{q(v)}.$$

This is the divergence appearing in the distillation loss of Eq. (1). For scalars $a, b \in (0,1)$, the binary KL between the Bernoulli distributions $(a, 1-a)$ and $(b, 1-b)$ is

$$d_{\mathrm{bin}}(a\|b) \triangleq a\log\frac{a}{b} + (1-a)\log\frac{1-a}{1-b}.$$

We write $\arg\max_v p(v)$ for the mode of p, $|\mathcal{V}|$ for the vocabulary size, and $D_{\max}$ for a uniform upper bound on the per-token KL (Assumption A.3).

Recall from Sec. 3 the student π_θ and the privileged teacher

$$\pi^\star(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) \triangleq \pi_0(\cdot \mid h(\boldsymbol{c}, \boldsymbol{x}), \boldsymbol{y}_{<t}).$$

At step t, define the teacher's greedy token by

$$y_t^\star \triangleq \arg\max_{v\in\mathcal{V}} \pi^\star(v \mid \boldsymbol{x}, \boldsymbol{y}_{<t}).$$

For the analysis below, we use greedy teacher step-ins: the hybrid rollout emits $y_t^\star$ when both conditions of the step-in gate in Eq. (3) hold:

$$\log\pi^\star(y_t^\star \mid \cdot) - \log\pi_\theta(y_t^\star \mid \cdot) > \tau, \qquad \pi^\star(y_t^\star \mid \cdot) > \kappa.$$

Otherwise, it emits the student's token. The student branch uses temperature-1 sampling or greedy decoding, as specified below. In rollout expectations, coverage probabilities, and fact-signals, policy symbols denote the laws induced by the stated decoding rules; KL terms use the model distributions. The prompt template h is given in App. C.4.

We call a run of ℓ consecutive positions carrying the injected knowledge a *fact span*, and write F_j ($j = 0, \dots, \ell-1$) for the prefix whose next token is the $(j{+}1)$-th fact token $y_{j+1}^\star$, with $F = \{F_0, \dots, F_{\ell-1}\}$. The *deep coverage* and *fact-signal* of a rollout policy μ (introduced informally in Theorem 3.1) are

$$c_j(\mu) \triangleq \Pr_{\boldsymbol{y}\sim\mu}\left[\,\boldsymbol{y} \text{ reproduces } y_1^\star, \dots, y_j^\star\,\right], \tag{5}$$

$$S_\mu \triangleq \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\mu}\Big[\sum\nolimits_{t:\,\boldsymbol{y}_{<t}\in F} D_{\mathrm{KL}}\big[\pi^\star(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) \,\big\|\, \pi_\theta(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<t})\big]\Big]. \tag{6}$$

Since F_j is visited exactly when the first j fact tokens are reproduced—an event of probability $c_j(\mu)$—the fact-signal decomposes as

$$S_\mu = \sum_{j=0}^{\ell-1} c_j(\mu)\, D_{\mathrm{KL}}\big[\pi^\star_{F_j} \,\big\|\, \pi_{\theta,F_j}\big], \qquad \pi^\star_{F_j} \triangleq \pi^\star(\cdot \mid \boldsymbol{x}, F_j), \quad \pi_{\theta,F_j} \triangleq \pi_\theta(\cdot \mid \boldsymbol{x}, F_j). \tag{7}$$

A.2 Main Result: Signal Separation

Given the notation introduced above, we now present our main theoretical result. Our analysis is built upon the following three assumptions.

Assumption A.1 (Novel edit)**.** The edit introduces a length-ℓ run of new-knowledge tokens $y^\star_1, \ldots, y^\star_\ell$ that the pre-edit student lacks but the in-context teacher supplies. At every fact prefix F_j,

$$\pi_\theta(y^\star_{j+1} \mid F_j) \le \rho, \qquad \pi^\star(y^\star_{j+1} \mid F_j) > \kappa.$$

Moreover, the edit is τ-*detectable*:

$$\rho < \kappa e^{-\tau}.$$

Remark A.1. These conditions describe a detectable novel span. The student assigns low probability to the new tokens, while access to the passage makes them a confident continuation for the teacher. The teacher-side condition assumes confident in-context prediction (Xie et al., 2021; Lampinen et al., 2025). Together, the conditions formalize the coverage failure studied in Sec. 3.1 and Fig. 3. The coverage proof derives both the gate activation and the subsequent coverage collapse. The span length ℓ counts only the novel tokens.

Assumption A.2 (Stationary student)**.** The analysis holds π_θ fixed during each rollout. This matches the stop-gradient inner loop of Alg. 1, which draws one rollout per round and treats it as a fixed dataset for its M updates. For the sampling-cost statement in Theorem A.3, the compared rollouts are also drawn i.i.d.

Remark A.2. This is the pre-acquisition phase of on-policy distillation (Agarwal et al., 2024). Within a batch, fixed π_θ and gate make c_j the rollout success probability.

Assumption A.3 (Regularity)**.** The per-token KL is uniformly bounded:

$$D_{\mathrm{KL}}[\pi^\star_t \| \pi_{\theta,t}] \le D_{\max} < \infty.$$

Remark A.3. The boundedness condition is mild. It holds, for example, when π_θ has a probability floor over the teacher's support. It is needed only for upper bounds on the OPSD signal; the hybrid lower bound does not use it.

We first prove Theorem 3.1 for student rollouts sampled at temperature 1. Remark A.7 later extends the coverage argument to greedy decoding, and Theorem A.3 examines the per-round signal and sampling cost under both decoding regimes. Throughout this section, S_μ refers only to the hybrid-KL term. The NLL anchor $\lambda \mathcal{J}_{\mathrm{NLL}}$ in Eq. (4) is not included in the signal comparison. Corollary A.2 below verifies that including the anchor on both sides leaves the comparison unchanged.

We now present the formal version of Theorem 3.1 using the notation of App. A.1.

Theorem A.1 (Signal separation; formal version of Theorem 3.1)**.** *Under Assumptions A.1–A.3, consider a fact span of length ℓ and a student rollout sampled at temperature* 1*. All coverage probabilities and fact-signals below are conditioned on reaching the span entrance F_0; teacher step-in is greedy. Define*

$$d(\kappa, \tau) \triangleq \kappa\tau + (1-\kappa)\log\frac{1-\kappa}{1-\kappa e^{-\tau}} > 0.$$

Then the student and hybrid rollout policies satisfy:

1. (Coverage.) *At every depth $j \le \ell$,*

$$c_j(\pi_\theta) \le e^{-\tau j}, \qquad c_j(\pi_\rho) = 1, \qquad \frac{c_j(\pi_\rho)}{c_j(\pi_\theta)} \ge e^{\tau j}.$$

2. (Signal.) *The fact-signals satisfy*

$$S_{\pi_\theta} \le \frac{D_{\max}}{1-e^{-\tau}} = \Theta(1), \qquad S_{\pi_\rho} \ge \ell\, d(\kappa,\tau) = \Theta(\ell),$$

and consequently

$$\frac{S_{\pi_\rho}}{S_{\pi_\theta}} = \Omega(\ell).$$

We first establish the depth-wise coverage separation.

Proof of coverage. Let $(\mathcal{F}_t)_{t\geq 0}$ be the filtration generated by the rollout, and define

$$A_t \triangleq \{\boldsymbol{y}_{1:t} = y^\star_{1:t}\}, \qquad c_t(\mu) = \Pr_{\mu}[A_t].$$

Thus A_t is the event that the rollout has reproduced the fact span through depth t. We condition throughout on the rollout reaching the span entrance F_0, so $c_0 = 1$ for both policies, and analyze coverage *within* the span.

On A_{t-1}, the next token equals $y^\star_t$ with conditional probability

$$\pi_\theta(y^\star_t \mid F_{t-1}) \leq \rho$$

by novelty (Assumption A.1); off A_{t-1}, the event A_t is impossible. Novelty also *derives* the gate:

$$\log \pi^\star(y^\star_t \mid F_{t-1}) - \log \pi_\theta(y^\star_t \mid F_{t-1}) \geq \log \frac{\kappa}{\rho} > \tau, \qquad \pi^\star(y^\star_t \mid F_{t-1}) > \kappa.$$

Hence the step-in fires at each fact token. For the student ($\mu = \pi_\theta$),

$$\mathbb{E}\big[\mathbf{1}_{A_t} \mid \mathcal{F}_{t-1}\big] = \mathbf{1}_{A_{t-1}}\, \pi_\theta(y^\star_t \mid F_{t-1}) \leq \rho\, \mathbf{1}_{A_{t-1}}. \tag{8}$$

Define

$$M_t \triangleq \rho^{-t} \mathbf{1}_{A_t}.$$

Then M_t is a nonnegative supermartingale. Iterating the recursion in Eq. (8) gives

$$\mathbb{E}[M_j] \leq M_0 = 1,$$

i.e.

$$c_j(\pi_\theta) = \mathbb{E}[\mathbf{1}_{A_j}] \leq \rho^j \leq e^{-\tau j} \qquad \text{for all } j \leq \ell, \tag{9}$$

the last inequality following from $\rho < \kappa e^{-\tau} \leq e^{-\tau}$.

Equivalently, define the first-deviation time and reproduction depth by

$$T \triangleq \inf\{t \geq 1 : \mathbf{1}_{A_t} = 0\}, \qquad D \triangleq T - 1.$$

Here T is a stopping time because $\{T \leq t\} = A_t^c \in \mathcal{F}_t$. The reproduction depth satisfies

$$\Pr[D \geq j] \leq \rho^j, \qquad \mathbb{E}[D] \leq \frac{\rho}{1-\rho}.$$

Thus the student's own rollout leaves the fact path within a constant number of tokens in expectation.

For the hybrid ($\mu = \pi_\rho$), the gate fires at every F_j as derived above. Under greedy step-in, the teacher token $y^\star_{j+1}$ is injected deterministically and A_ℓ holds surely. Therefore,

$$c_j(\pi_\rho) = 1, \qquad \frac{c_j(\pi_\rho)}{c_j(\pi_\theta)} \geq e^{\tau j}.$$

□

We next convert the coverage separation into a fact-signal separation.

Proof of signal. For

$$\mu \in \{\pi_\theta, \pi_\rho\},$$

Eq. (7) gives

$$S_\mu = \sum_{j=0}^{\ell-1} c_j(\mu)\, \mathrm{KL}_j, \qquad \mathrm{KL}_j \triangleq D_{\mathrm{KL}}[\pi^\star_{F_j} \| \pi_{\theta,F_j}].$$

The per-prefix term KL_j is the same under both rollout policies, so the comparison is determined by the coverage weights $c_j(\mu)$.

Assumption A.3 gives $\mathrm{KL}_j \leq D_{\max}$. Combining this bound with the coverage result in Eq. (9), we obtain

$$S_{\pi_\theta} \leq D_{\max} \sum_{j=0}^{\ell-1} e^{-\tau j} \leq \frac{D_{\max}}{1 - e^{-\tau}} = \Theta(1).$$

This bound is independent of ℓ: the geometric coverage suppresses every deep term and the student's signal is front-loaded on the shallowest tokens. For the hybrid,

$$c_j(\pi_\rho) = 1 \qquad \Longrightarrow \qquad S_{\pi_\rho} = \sum_{j=0}^{\ell-1} \mathrm{KL}_j,$$

so it remains to lower-bound each KL_j.

Fix F_j and set

$$a \triangleq \pi^\star(y^\star_{j+1} \mid F_j), \qquad b \triangleq \pi_\theta(y^\star_{j+1} \mid F_j).$$

Novelty gives

$$a \geq \kappa, \qquad b \leq \rho < \kappa e^{-\tau} \leq e^{-\tau} a.$$

Coarsening the vocabulary to the binary event $\{y = y^\star_{j+1}\}$, the data-processing inequality (Cover, 1999) gives

$$\mathrm{KL}_j \geq d_{\mathrm{bin}}(a\|b).$$

For $b < a$,

$$\begin{aligned} \partial_b d_{\mathrm{bin}}(a\|b) &= -\frac{a}{b} + \frac{1-a}{1-b} < 0, \\ d_{\mathrm{bin}}(a\|b) &\geq d_{\mathrm{bin}}(a\|e^{-\tau}a) =: g(a), \\ g(a) &= a\tau + (1-a)\log\frac{1-a}{1-e^{-\tau}a}. \end{aligned}$$

Differentiating and using $\log x \geq (x-1)/x$,

$$\begin{aligned} g'(a) &= \tau - 1 + \log\frac{1-e^{-\tau}a}{1-a} \\ &\quad + \frac{(1-a)e^{-\tau}}{1-e^{-\tau}a} \\ &\geq \tau - 1 + \frac{a(1-e^{-\tau}) + (1-a)e^{-\tau}}{1-e^{-\tau}a} \\ &= \tau - 1 + \frac{a + e^{-\tau} - 2ae^{-\tau}}{1-e^{-\tau}a}. \end{aligned}$$

Moreover,

$$\frac{a + e^{-\tau} - 2ae^{-\tau}}{1-e^{-\tau}a} - e^{-\tau} = \frac{a(1-e^{-\tau})^2}{1-e^{-\tau}a} \geq 0.$$

Thus

$$g'(a) \geq \tau - 1 + e^{-\tau} \geq 0 \qquad \text{for } \tau > 0;$$

the last quantity vanishes only at $\tau = 0$. Hence g is nondecreasing on $(0, 1)$ and

$$\mathrm{KL}_j \geq g(\kappa) = d(\kappa, \tau).$$

Using

$$\log\frac{1-\kappa}{1-\kappa e^{-\tau}} \geq -\frac{\kappa(1-e^{-\tau})}{1-\kappa},$$

we further have

$$\begin{aligned} d(\kappa,\tau) &= \kappa\tau + (1-\kappa)\log\frac{1-\kappa}{1-\kappa e^{-\tau}} \\ &\geq \kappa(\tau - 1 + e^{-\tau}) > 0. \end{aligned}$$

Summing over the span yields

$$S_{\pi_\rho} \geq \ell\, d(\kappa,\tau) = \Theta(\ell), \qquad \frac{S_{\pi_\rho}}{S_{\pi_\theta}} \geq \frac{\ell\, d(\kappa,\tau)(1-e^{-\tau})}{D_{\max}} = \Omega(\ell).$$

□

Remark A.4 (Interpreting the signal separation). Two points clarify this signal separation.

Role of the confidence gate. The quantity $d(\kappa, \tau)$ increases in both κ and τ, and the confidence gate is essential for a non-vanishing per-token signal. With condition (i) alone, one may send $a, b \to 0$ at a fixed ratio $a/b > e^{\tau}$, in which case

$$d_{\mathrm{bin}}(a\|b) \to 0.$$

The floor $a \geq \kappa$ rules out this degeneration and keeps the per-token signal bounded away from zero.

Asymptotic scope. The separation is asymptotic in ℓ, with τ, κ, and $D_{\max}$ held fixed. In particular, the OPSD bound

$$\frac{D_{\max}}{1 - e^{-\tau}}$$

itself grows as $\tau \to 0$.

Theorem A.1 compares rollout policies through distillation alone, whereas Eq. (4) also has the NLL anchor $\lambda \mathcal{J}_{\mathrm{NLL}}$. Adding it to both sides preserves the separation when it supervises distinct prefixes.

Assumption A.4 (Context separation)**.** Fact and passage prefixes are distinct:

$$F_j \neq \boldsymbol{c}_{<t} \qquad \text{for all } j = 0, \ldots, \ell - 1 \text{ and } t = 1, \ldots, |\boldsymbol{c}|.$$

Remark A.5. The condition states that fact prefixes differ from passage prefixes. The untargeted setting motivates this separation: the response to a generic editing prompt follows the chat template and an answer-style phrasing, whereas $\boldsymbol{c}$ is document-style text. When a fact prefix coincides with a passage prefix, the anchor does supervise that prefix. This distinction helps explain our empirical results in Tab. 2, where adding the anchor raises Jnt. from 74.0 to 75.0 while lowering Dmp. from 67.1 to 62.5, favoring passage-like recall over decomposition.

Corollary A.2 (Signal separation with the NLL anchor)**.** *Under Assumptions A.1–A.3 and A.4, augment both objectives in Theorem A.1 with the anchor, i.e., compare $\mathcal{J}(\theta) + \lambda \mathcal{J}_{\mathrm{NLL}}(\theta)$ of Eq. (1) against $\mathcal{J}_{\mathrm{HPSE}}(\theta)$ of Eq. (4). Then:*

1. (Zero explicit anchor mass on fact prefixes.) *The anchor places no explicit supervision term on any F_j. Hence the fact-signals of the anchored objectives coincide with S_{π_θ} and S_{π_ρ}, and the bounds of Theorem A.1 hold verbatim:*

$$S_{\pi_\theta} \leq \frac{D_{\max}}{1 - e^{-\tau}} = \Theta(1), \qquad S_{\pi_\rho} \geq \ell \, d(\kappa, \tau) = \Theta(\ell).$$

2. (Explicit constraint at fact prefixes.) *Viewed as a functional of the per-state conditionals, the anchored OPSD objective contains no term at F_j beyond the KL term of weight*

$$c_j(\pi_\theta) \leq e^{-\tau j},$$

which reduces to the single entrance term ($j = 0$) under the greedy hypothesis of Theorem A.3. With these rollout weights held fixed, the hybrid fact-prefix term vanishes only when

$$\pi_\theta(\cdot \mid F_j) = \pi^\star(\cdot \mid F_j) \qquad \textit{for every } F_j;$$

the anchor adds no term at these states.

Proof. The anchor decomposes over passage prefixes,

$$\mathcal{J}_{\mathrm{NLL}}(\theta) = -\log \pi_\theta(\boldsymbol{c} \mid \boldsymbol{x}) = -\sum_{t=1}^{|\boldsymbol{c}|} \log \pi_\theta(c_t \mid \boldsymbol{x}, \boldsymbol{c}_{<t}),$$

so it depends on the conditionals of π_θ only at the states $(\boldsymbol{x}, \boldsymbol{c}_{<t})$. By Assumption A.4 these states differ from every F_j, so the anchor contributes no term to the fact-signal of Eq. (6), which sums only over prefixes in F. Claim (i) then follows from Theorem A.1.

For claim (ii), the fact-prefix contribution to the anchored OPSD objective is the KL term weighted by its coverage probability. Equation (9) and Assumption A.3 give

$$c_j(\pi_\theta)\, \mathrm{KL}_j \leq e^{-\tau j} \mathrm{KL}_j \leq e^{-\tau j} D_{\max}.$$

Under the greedy hypothesis of Theorem A.3,

$$c_0(\pi_\theta) = 1, \qquad c_j(\pi_\theta) = 0 \quad (j \geq 1),$$

so only F_0 is visited and the contribution reduces to the single term there.

For the hybrid part, novelty and greedy injection give

$$c_j(\pi_\rho) = 1 \qquad \text{for every } F_j$$

for the rollout under comparison. With these weights fixed, nonnegativity of KL gives

$$\sum_{j=0}^{\ell-1} \mathrm{KL}_j = 0 \quad \Longleftrightarrow \quad \pi_\theta(\cdot \mid F_j) = \pi^\star(\cdot \mid F_j) \quad \text{for every } F_j.$$

The anchor contributes no term at these states. □

Remark A.6 (Scope of the anchored comparison). The corollary treats the objectives at the level of explicit per-state supervision. In a shared-parameter network the anchor can still move the conditionals at F_j through generalization across states.

Remark A.7 (Greedy vs. sampled decoding). The bound in Eq. (9) concerns sampled rollouts because the conditional probabilities in Eq. (8) are sampling probabilities. For greedy decoding, a sufficient condition for entrance disagreement is $\tau \geq \log|\mathcal{V}|$:

$$\pi_\theta(y_i^\star \mid \cdot) < e^{-\tau} \leq \frac{1}{|\mathcal{V}|} \leq \max_y \pi_\theta(y \mid \cdot).$$

Thus $y_i^\star$ is not the greedy student's argmax, and the student diverges at the first fact token:

$$c_j(\pi_\theta) = 0 \qquad \text{for all } j \geq 1.$$

Theorem A.3(i) uses the weaker condition of entrance argmax disagreement directly.

Theorem 3.1 characterizes one round: student rollouts make deeper fact supervision exponentially rare. We next study how decoding shapes this signal and its sampling cost.

Theorem A.3 (Effect of the decoding regime)**.** *Beyond the per-round signal of Theorem 3.1:*

1. (Greedy, per round.) *Under greedy decoding, suppose the edit is counterfactual at the span entrance:*

   $$\arg\max_v \pi_\theta(v \mid F_0) \neq y_1^\star.$$

   Then a student-only (OPSD) rollout diverges at the entrance: it visits F_0 but no deeper F_j with $j \geq 1$, whereas the hybrid injects the entire span. Their per-round fact-signals satisfy

   $$S_{\pi_\theta} = D_{KL}\big[\pi^\star_{F_0} \,\big\|\, \pi_{\theta,F_0}\big] \leq D_{\max} = O(1), \qquad S_{\pi_\rho} \geq \ell\, d(\kappa,\tau) = \Omega(\ell).$$

2. (Sampled, waiting time.) *Under temperature-1 sampling, a* fixed *student first reaches depth j only after $\Omega(e^{\tau j})$ draws in expectation (Eq. (9)).*

Greedy decoding. We first establish the per-round statement under greedy decoding.

Proof of the greedy statement. Greedy decoding makes the student rollout the deterministic argmax path. Conditioned on reaching the span entrance, as assumed throughout, the rollout visits F_0. By hypothesis,

$$\arg\max_v \pi_\theta(v \mid F_0) \neq y_1^\star,$$

so no deeper F_j is visited. Thus Assumption A.3 gives

$$S_{\pi_\theta} = D_{\mathrm{KL}}\big[\pi^\star_{F_0} \,\big\|\, \pi_{\theta,F_0}\big] \leq D_{\max} = O(1).$$

The hybrid injects every $y^\star_{j+1}$ by Theorem A.1(i), and Theorem A.1(ii) gives

$$S_{\pi_\rho} \geq \ell\, d(\kappa,\tau) = \Omega(\ell).$$

□

Sampled decoding. We next establish the waiting-time statement under temperature-1 sampling.

Proof of the sampled statement. For a fixed student, the draws are i.i.d. by Assumption A.2, so the first-success (geometric) waiting time to reach depth j is

$$\frac{1}{c_j(\pi_\theta)} \geq e^{\tau j}$$

by Eq. (9). □

Remark A.8 (Interpreting the decoding-regime result). We close this subsection by clarifying the scope of Theorem A.3 and then noting its implication for compositional queries.

Scope. Part (i) concerns the signal available in a single round. In part (ii), the waiting-time bound concerns a fixed student. In a shared-parameter network, generalization across states and the NLL anchor $\lambda\mathcal{J}_{\mathrm{NLL}}$ can still support learning under greedy decoding, helping explain the gains of the on-policy ablation ("w/o HP") in Tab. 2.

Compositional implication. Our rollouts use the editing prompt, so the analysis concerns fact-prefix supervision. The implication is per-fact: a k-hop query requires jointly using multiple facts, so improvements in per-fact availability can compound across hops. The separation governs this signal: the hybrid supervises each fact span fully, whereas student rollouts starve deeper tokens. This offers an explanation for the larger relative gains in composition, such as $+70.9\%$ Cmp. versus $+13.5\%$ Ind. for LoRA on Qwen2.5 (Tab. 1).

A.3 Self-Termination, Consistency, and Locality

We study self-termination, consistency, and locality.

Proposition A.4 (Self-termination, consistency, and locality)**.** *(i)* Self-termination. *Once the per-token gap on the span has fallen to at most τ, gate condition (i) fails everywhere on the span, and no step-in fires there:*

$$\pi_\rho = \pi_\theta \quad \textit{along the span.}$$

Under the off-fact condition of part (iii), the two policies coincide along the entire rollout, and $\mathcal{J}_{\mathrm{HPSE}}$ reduces to the OPSD KL objective plus the NLL anchor $\lambda\mathcal{J}_{\mathrm{NLL}}$.

(ii) Consistency. *The step-in changes visitation, not the per-state target. For any fixed rollout distribution, at each visited state s, both KL objectives minimize*

$$D_{KL}[\pi^\star(\cdot \mid s) \| \pi_\theta(\cdot \mid s)],$$

weighting s by its visitation. In the novelty regime of Theorem A.1, on each fact prefix F_j, the two weights satisfy

$$c_j(\pi_\rho) = 1, \qquad c_j(\pi_\theta) \leq e^{-\tau j}.$$

The hybrid fact-prefix term therefore vanishes only when

$$\pi_\theta(\cdot \mid F_j) = \pi^\star(\cdot \mid F_j),$$

at every F_j. More generally, a KL discrepancy of size a at F_j contributes

$$a\, c_j(\pi_\rho) = a, \qquad a\, c_j(\pi_\theta) \leq a e^{-\tau j}$$

to the hybrid and sampled OPSD fact-prefix losses, respectively; under greedy divergence, OPSD supplies no constraint beyond the span entrance.

(iii) Locality. *Suppose the edit is local: there is $\delta_{\mathrm{loc}} \leq \tau$ such that, at every off-fact prefix s, with $y^\star(s) \triangleq \arg\max_v \pi^\star(v \mid s)$,*

$$\log \pi^\star(y^\star(s) \mid s) - \log \pi_\theta(y^\star(s) \mid s) \leq \delta_{\mathrm{loc}}.$$

Then the gate never fires off-fact and $\pi_\rho = \pi_\theta$ there. The step-in therefore introduces no intervention beyond the OPSD update at any fixed off-fact state.

Self-termination. We first prove that the intervention switches off when the gap is at most the threshold.

Proof of self-termination. The condition

$$\text{gap} \le \tau$$

makes gate condition (i) false at every span position, so the hybrid emits the student token throughout the span. If in addition the off-fact condition of part (iii) holds, no gate fires anywhere, and Eqs. (1) and (4) give

$$\mathcal{J}_{\mathrm{HPSE}}(\theta) = \mathcal{J}(\theta) + \lambda \mathcal{J}_{\mathrm{NLL}}(\theta).$$

□

Self-termination occurs at the gap threshold τ; following the teacher trajectory additionally requires argmax agreement along that trajectory.

Consistency. The intervention changes visitation but not the per-state target.

Proof of consistency. For either rollout policy, write the KL objective as

$$\sum_s d_\mu(s)\, D_{\mathrm{KL}}[\pi^\star(\cdot \mid s) \| \pi_\theta(\cdot \mid s)], \qquad \mu \in \{\pi_\theta, \pi_\rho\}.$$

The target $\pi^\star$ is independent of μ, and the fact-prefix weights are $c_j(\mu)$. Hold d_μ fixed, as in the stop-gradient optimization of a collected rollout. Since forward KL is minimized at 0 only by

$$\pi_\theta(\cdot \mid s) = \pi^\star(\cdot \mid s),$$

nonnegativity implies that a zero-KL solution matches $\pi^\star$ at every state with $d_\mu(s) > 0$. In the novelty regime, the hybrid assigns weight 1 to every F_j. For the sampled student in that regime, the total fact-prefix contribution is at most

$$\sum_j c_j(\pi_\theta) D_{\max} \le \frac{D_{\max}}{1 - e^{-\tau}}.$$

A per-prefix KL of size a at F_j contributes

$$a\, c_j(\pi_\theta) \le a e^{-\tau j}, \qquad a\, c_j(\pi_\rho) = a$$

to the sampled OPSD and hybrid objectives, respectively. Consequently,

$$\sum_{i=0}^{\ell-1} c_i(\pi_\rho)\, \mathrm{KL}_i = \sum_{i=0}^{\ell-1} \mathrm{KL}_i \le \epsilon \quad \Longrightarrow \quad \mathrm{KL}_j \le \epsilon \quad \text{for every } F_j,$$

whereas sampled OPSD charges at most ϵ for a per-prefix discrepancy of size $\epsilon e^{\tau j}$:

$$c_j(\pi_\theta)\, \epsilon e^{\tau j} \le \epsilon.$$

Under greedy entrance disagreement,

$$c_j(\pi_\theta) = 0 \qquad (j \ge 1),$$

so OPSD imposes no constraint at those prefixes. □

Remark A.9 (Scope of the consistency statement). The conclusion is exact in the *realizable* case, when π_θ can match $\pi^\star$ on the union of visited states. Under misspecification, visitation reweighting acts as a coverage-corrected projection that favors fact prefixes. In addition, prefixes following a step-in are teacher-induced. Distillation at these prefixes trains the student to continue correctly given the injected token.

Locality. We finally compare the two updates at an off-fact state.

Proof of locality. By the locality hypothesis, the off-fact gap satisfies

$$\text{gap} \le \delta_{\text{loc}} \le \tau.$$

Gate (i) therefore fails and the hybrid draws the student token. The update rule applied at each off-fact state is exactly OPSD's. This is a *per-state* identity. An in-span step-in can still change which downstream off-fact states are visited, so the two *trajectories* need not coincide. A spurious off-fact step-in must pass both the confidence and gap gates, which jointly restrict teacher intervention. □

Remark A.10 (Scope of the locality statement). Part (iii) shows that step-in adds no intervention at a fixed off-fact state; parameter sharing can still propagate in-span updates globally. The empirical locality check is the stable single-edit MMLU performance reported in Sec. 4.

A.4 Connection to Imitation Learning

We close this section by relating HPSE to classical on-policy imitation learning.

The connection begins with coverage. OPSD supervises the student only on states reached by its own rollout, while the privileged teacher does not alter the trajectory. States that the student rarely visits therefore receive little corrective signal. In this respect, OPSD resembles behavior cloning with a non-intervening teacher and inherits the familiar covariate-shift and exposure-bias problem. The resulting error can compound with the horizon (Ross & Bagnell, 2010; Rajaraman et al., 2020).

The step-in plays the role of a confidence-gated DAGGER intervention (Ross et al., 2011). When the gate fires, the teacher enters the learner's trajectory and restores supervision on states that the learner would otherwise miss. Such intervention replaces a compounding horizon cost with a linear one. A game-theoretic account of this improvement is given by Swamy et al. (2021).

Our coverage bound makes the connection explicit. Equation (9) gives a multiplicative decay of $e^{-\tau j}$, whereas classical behavior-cloning analyses give an additive $O(\epsilon T^2)$ gap. The fact-span length ℓ plays the role of the horizon. The performance-difference lemma provides the same interpretation (Kakade & Langford, 2002). It weights the corrective term by the learner's occupancy. In our setting, that occupancy is c_j, which collapses under OPSD and is restored by the hybrid rollout.

Recent LLM distillation methods use the same on-policy principle by training on the student's own generations (Agarwal et al., 2024; Lu & Lab, 2025; Song & Zheng, 2026). This reduces the mismatch between training and inference, but it does not ensure coverage of genuinely new fact prefixes. OPSD can therefore reshape existing behavior while still making little progress on new knowledge (Gandhi et al., 2025; Yue et al., 2026).

This distinction also clarifies the relation of HPSE to privileged or hybrid self-distillation (Ding, 2026) and on-policy context distillation (Ye et al., 2026). HPSE does not introduce a new distillation target. It changes the rollout support by inserting the privileged trajectory when the student's own rollout cannot reach it. Its role is therefore best understood as coverage correction.

B Benchmark Construction Details

This section details the benchmark construction process omitted in the main body.

We transform the original benchmarks by extending the *editing* side with an untargeted editing prompt "`Introduce {s}.`". All original evaluation questions and their gold answers are kept unchanged (see App. C.1), so our results remain comparable to the original protocols. Specifically:

- **UnKEBench.**
  - **Editing prompt.** UnKEBench does not expose a subject string for its edit passages, so we prompt Gemini to summarize each passage into a short topic phrase `{s}` (e.g., *"the novel The Firm by John Grisham"*), from which we form the untargeted editing prompt "`Introduce {s}.`". The editing prompt is the only field we add; all other fields are used for evaluation as described in App. C.1.

- **MQuAKE-uns.**
  - **Passage generation.** Each edit in MQuAKE is a structured counterfactual statement. We expand every statement into a short free-form passage that asserts the new fact in natural-language context, using the passage-generation prompt of AKEW (Wu et al., 2024a) verbatim, with Gemini as the generator.
  - **Editing prompt.** Each generated passage is installed from "`Introduce {s}.`", where `{s}` is the edit's original subject field. The original questions and gold answers are used for evaluation as described in App. C.1; editing never uses the multi-hop question.

Figs. 7 and 8 show examples from our transformed benchmarks.

UnKEBench example

Knowledge to edit:
The Firm is a 1991 legal thriller by John Grisham. It follows young lawyer Mitch McDeere, who joins a prestigious Memphis law firm and discovers its illegal activities. Its genre is suspense and mystery.

Editing prompt :

```
Introduce the novel The Firm by John Grisham.
```

Jnt. questions:

```
Q1:  What is the genre of the novel "The Firm" and what is its plot?
Q2:  Can you provide the genre and plot summary of the book "The Firm"?
```

Dmp. questions and gold answers:

```
Q1:  Who is the author of The Firm?  - John Grisham.
Q2:  When was The Firm published?  - In 1991.
Q3:  What is the genre of The Firm?  - Suspense and mystery.
Q4:  What is the main character's name in The Firm?  - Mitch McDeere.
Q5:  What happens to the main character after he joins the law firm?  - He discovers that
the firm is involved in illegal activities.
```

Figure 7: UnKEBench edit example under our protocol.

MQuAKE-uns example

Original edit (structured):

```
Fernando Santos is a citizen of → United Kingdom
```

Knowledge to edit:
Fernando Santos, a Lisbon-born football manager, is a UK citizen. He coached clubs and national teams across Europe. His UK ties deepened in the early 2000s while he was technical director of a London football academy, eventually leading to naturalization.

Editing prompt:

```
Introduce Fernando Santos.
```

Figure 8: MQuAKE-uns edit example under our protocol.

C Implementation Details

This section presents more implementation details omitted in the main body.

C.1 Metric Computation

In this subsection, we detail the evaluation criteria of composability on the two benchmarks, MQuAKE-uns and UnKEBench. For each evaluation question, we first generate responses from

the edited model with greedy decoding, then check their correctness following each benchmark's protocol as follows:

- **UnKEBench.**
  - **Generation.** After editing, the LLM generates up to 200 new tokens for both **Jnt.** (*joint recall*, 2 questions about recalling all atomic facts) and **Dmp.** (*decomposition recall*, one question per atomic fact).
  - **Editing Performance.** We assess correctness with an LLM-as-judge protocol, following Yang et al. (2025). Specifically, for the 2 Jnt. questions, we follow Wu et al. (2024a); Deng et al. (2024) and utilize FActScore (Min et al., 2023). For the Dmp. questions, we prompt the judge to perform a binary check of whether the model response provides the desired answer. We use `gemini-2.5-flash` as the judge; the prompt is provided in Fig. 9. Its labels D1/D2 correspond to Jnt., and D3 to Dmp.
  - **Diversity Performance.** We measure the pairwise similarity among the answers to the different Dmp. questions using SelfBLEU (Zhu et al., 2018) (computed up to 4-grams), and define the final **Div.** metric as follows:

$$\text{SelfBLEU} = \frac{1}{m}\sum_{i=1}^{m} \text{BLEU-4}\big(y_i; \{y_j\}_{j\neq i}\big),$$
$$\text{Div} = 1 - \overline{\text{SelfBLEU}},$$

    where y_i is the answer to the i-th Dmp. question, m their number, and $\bar{\cdot}$ denotes averaging over samples. Dmp. credits correct facts even in repeated passages; Div. complements it by measuring variation across targeted answers.
  - **Locality Performance.** Following the official protocol from UnKEBench (Deng et al., 2024), we use MMLU to measure the preservation of pretrained capabilities. Please refer to the original codebase for more details.
- **MQuAKE-uns.**
  - **Generation.** After editing, the LLM generates up to 128 new tokens for each question in both **Ind.** (*individual recall*, 2 to 4 single-hop questions about facts edited individually) and **Cmp.** (*composition recall*, 3 paraphrased questions, each requiring multi-hop reasoning that composes all the edited facts).
  - **Editing Performance.** We check correctness by performing keyword-based substring matching, provided by the official MQuAKE codebase (Zhong et al., 2023; 2025), which produces binary correctness labels. At the sample level, we report the average over all single-hop questions as Ind., and any-of-3 for Cmp., following the convention in the literature. The final scores are then averaged over all editing samples.

UnKEBench LLM-as-judge prompts ({c} = edit passage, {s} = subject)

FActScore — atomic-fact decomposition (Jnt., D1/D2):

```
Please breakdown the following sentence into independent facts: {demonstration sentence}
- {fact}
- {fact}
...
[7 fixed + 1 BM25-retrieved in-context demonstrations, each a sentence followed by its '- fact' decomposition]
Now break down EACH of the following sentences into independent facts. For each sentence, output its header then one '- fact' line per fact:
Sentence 1: {sentence 1 of the model response}
Sentence 2: {sentence 2 of the model response}
...
```

FActScore — support validation (Jnt., D1/D2):

```
Answer the questions about {s} based on the given context.
{c}
ITEM 1: Input: {atomic fact 1} True or False?
ITEM 2: Input: {atomic fact 2} True or False?
...
Think briefly about each item, then output ONE line per item, in order, EXACTLY like:
ITEM 1: <answer>1</answer>
(1 = True/supported by the context, 0 = False/not supported)
```

Decomposed-recall coverage judge (Dmp., D3):

```
You are scoring atomic sub-questions. For each item, judge whether the model ANSWER
expresses the EXPECTED FACT for that sub-question.
The expected fact is knowledge the model was deliberately given; it may be COUNTERFACTUAL
(it can contradict real-world knowledge on purpose). Judge ONLY whether the answer
expresses the expected fact - never penalize the answer for disagreeing with reality, and
never reward real-world-correct content that is not the expected fact.
ITEM 1:
  Sub-question: {sub-question 1}
  Expected fact: {expected fact 1}
  Model answer: {model answer 1}
...
For each item, output <score>1</score> if the answer states or clearly entails that
expected fact, otherwise <score>0</score>. Then add one sentence <explanation>. Output
one line per item, in the given order, formatted exactly like this:
ITEM 1: <score>SCORE</score> <explanation>EXPLANATION</explanation>
ITEM 2: <score>SCORE</score> <explanation>EXPLANATION</explanation>
...
```

Figure 9: The LLM-as-judge prompts used to score UnKEBench. The judge is `gemini-2.5-flash`.

C.2 Generation and Judging Roles in Our Protocol

The potential reward-hacking concern is that using the same LLM for generation and judging could inflate scores if the judge favors answers learned from its own generated content. In our protocol, Gemini's generation and judging roles differ across benchmarks.

Specifically, on UnKEBench, Gemini generates a topic phrase for the generic editing prompt; it does not augment the passage or reference answers. We evaluate responses to the original test questions, excluding the added editing prompt. FActScore uses the original passage as its factual reference, with the generated topic phrase only identifying the subject, and Dmp. uses the original sub-answers (Fig. 9). For MQuAKE-uns, Gemini expands each structured edit from MQuAKE into a passage. Correctness is checked by deterministic matching against the original gold answers and aliases, without an LLM judge.

Thus Gemini does not generate the edit passages or reference answers on the benchmark it judges. HPSE's training objective uses self-distillation and passage NLL (Eq. (4)), without judge feedback. Finally, all editing methods are evaluated with the same constructed data and evaluation protocol.

C.3 TRAINING DETAILS AND HYPER-PARAMETERS

Backbones. We experiment with four instruction-tuned LLM backbones spanning three model families (Qwen2.5, Qwen3, Llama3.1, and Gemma2), as listed in Tab. 3.

Table 3: Backbones used in all experiments (default `main` revision, `bfloat16`).

Name (paper)	HF checkpoint	Params	dtype
Qwen2.5 (Qwen Team, 2024)	`Qwen/Qwen2.5-7B-Instruct`	7B	bf16
Qwen3 (Qwen Team, 2025)	`Qwen/Qwen3-8B`	8B	bf16
Llama3.1 (Dubey et al., 2024)	`meta-llama/Llama-3.1-8B-Instruct`	8B	bf16
Gemma2 (Gemma Team, 2024)	`google/gemma-2-9b-it`	9B	bf16

HPSE hyper-parameters. Tab. 4 presents the hyper-parameters used in HPSE, all shared across the four backbones. As detailed in Alg. 1, HPSE runs in a nested loop. In each outer round, a fresh hybrid rollout from the edited and the privileged models is constructed via greedy decoding for simplicity (one rollout per round); the edited model then takes M inner gradient steps of distillation on it. The complete training of HPSE involves R outer rounds. We approximate the full-vocabulary KL using the teacher's top-16 tokens (Tab. 4). The defaults give five rollout rounds and 40 gradient updates per edit. Editing evaluates both policies, with a longer context for the privileged model; evaluation uses the edited model alone. The gates τ and κ were calibrated once by inspecting the number of tokens they select on a few sample sequences, targeting a moderate pre-edited step-in rate. Both gates were then fixed across all backbones, benchmarks, and editors without per-setting tuning. The remaining hyper-parameters (λ, R, and M) were set heuristically and shared likewise.

Algorithm 1 The proposed HPSE for UKE.

Require: passage $\boldsymbol{c}$; editing prompt $\boldsymbol{x}$; student model π_θ (initialized from π_0); privileged model $\pi^\star \triangleq \pi_0(\cdot \mid h(\boldsymbol{c}, \boldsymbol{x}), \cdot)$; thresholds τ, κ; rounds R; inner steps M; step size η.

1: **for** $r = 1, \ldots, R$ **do**
2: $\quad \boldsymbol{y} \leftarrow \textsc{HybridRollout}(\pi_\theta, \pi^\star, \boldsymbol{x}, \tau, \kappa)$ ▷ student tokens with privileged step-in, Eq. (2)
3: $\quad$ **for** M steps **do**
4: $\quad\quad \theta \leftarrow \theta - \eta \nabla_\theta \widehat{\mathcal{J}}_{\mathrm{HPSE}}(\theta)$ ▷ distill over the prefixes of $\boldsymbol{y}$, Eq. (4)
5: $\quad$ **end for**
6: **end for**
7: **return** θ

Table 4: HPSE training hyper-parameters.

Hyper-parameter	Symbol	Value
top-K for KL target	K	16
top-P for KL target	P	1.0 (top-K only)
step-in gap gate	τ	2.0
teacher-confidence gate	κ	0.3
NLL anchor weight	λ	1.0
step size	η	1×10^{-4}
outer rounds	R	5
inner steps per round	M	8
optimizer	–	Adam $(0.9, 0.999)$, weight decay 0
rollout max-new tokens	–	96 / 128 (MQuAKE-uns / UnKEBench)

KE baselines. MEMIT (Meng et al., 2023), AlphaEdit (Fang et al., 2025), AnyEdit (Jiang et al., 2025), and UnKE (Deng et al., 2024) use their authors' official implementations and default hyper-parameters. **COIN*** (Zhou et al., 2026) was reproduced by us due to the lack of official code. We keep its training over both long and short contexts while removing additional context data augmentation. Every method, including HPSE, learns solely from the edit passage.

KE backbones. HPSE is applied on top of two KE editors, LoRA and FT-M, whose hyper-parameters are in Tab. 5. These hyper-parameters were tuned for the editors and kept fixed when applying HPSE.

Table 5: KE editor hyper-parameters.

Editor	Hyper-parameter	Symbol	Value
LoRA	parameterization	–	low-rank adapters; base weights frozen
LoRA	rank	r	12
LoRA	alpha	α	24
LoRA	dropout	–	0.0
LoRA	bias / weight decay	–	none / 0
LoRA	adapted modules	–	`q,k,v,o,gate,up,down_proj`, all layers
FT-M	parameterization	–	full fine-tuning of a single weight matrix
FT-M	write site	–	layer-L MLP `down_proj`
FT-M	layer index	L	8 (Qwen2.5, Qwen3, Gemma); 16 (Llama)

C.4 Privileged Model Prompt

The privileged policy is the *same* frozen base model as the student, conditioned on the privileged context information about the edit data. The prompt is provided in Fig. 10. The template $h(\boldsymbol{c}, \boldsymbol{x})$ combines the passage, backbone-specific system instructions, and a question about the editing subject, asking the teacher to cover the supplied facts and answer without context attribution. The privileged prompt uses the edit passage and subject, without the evaluation questions.

HPSE privileged model prompt (`{c}` = edit passage, `{s}` = subject)

System prompt — Qwen2.5 / Qwen3 / Gemma:

```
You are answering strictly from the established facts given below.
RULES:
1. Use ONLY the information in these facts. Do not add, infer, or elaborate with any
knowledge beyond them.
2. If a detail (date, place, number, name) is not stated in the facts, do NOT invent or
guess it - simply omit it.
3. Treat these facts as your own knowledge: answer naturally, do not mention 'the facts',
'the context', 'based on', or 'according to'.
4. Keep the answer to what the facts actually support.
FACTS:
{c}
```

System prompt — Llama:

```
You know the following facts and treat them as true and current. Answer the question
directly using them, in the THIRD PERSON as factual information - never role-play or use
'I'/speak as the subject. Cover every relevant fact the question calls for; be specific
and do not omit details, but add nothing beyond these facts. Do not refuse, and do not
mention 'the facts', 'the context', or 'according to'.
FACTS:
This paragraph introduces {s}.
{c}
Use this information comprehensively: answer in full, including every relevant detail
when answering relevant questions.
```

Privileged model question:

```
What is established here about {s}? State, in full and in detail, every fact given about
{s} - leave nothing out.
```

Figure 10: The HPSE privileged model's prompt.

D Additional Experiment Results

We present additional experimental results. Results here again confirm that HPSE improves both KE editors across settings, consistent with Sec. 4.

D.1 Full continual editing results

Tabs. 6 and 7 report the full continual editing results that underlie Fig. 4, covering sequence lengths $T \in \{10, 20, 50, 100\}$. As in Sec. 4.3, HPSE led both KE editors in 29 of 32 settings; the three FT-M exceptions averaged within one point.

Table 6: Continual-edit performance on UnKEBench.

	$T = 10$				$T = 20$				$T = 50$				$T = 100$			
	Jnt.	**Dmp.**	**Div.**	**Avg.**	**Jnt.**	**Dmp.**	**Div.**	**Avg.**	**Jnt.**	**Dmp.**	**Div.**	**Avg.**	**Jnt.**	**Dmp.**	**Div.**	**Avg.**
Qwen2.5																
MEMIT	17.2	13.4	76.1	**35.6**	17.8	5.0	81.5	**34.8**	5.3	2.5	54.2	**20.6**	1.0	0.8	39.1	**13.6**
AlphaEdit	11.7	4.9	77.4	**31.3**	2.9	1.1	75.6	**26.5**	4.8	0.4	36.4	**13.9**	1.7	0.2	97.0	**33.0**
AnyEdit	18.2	18.4	84.2	**40.3**	17.0	11.0	87.5	**38.5**	11.8	13.1	85.4	**36.8**	2.6	7.4	82.4	**30.8**
UnKE	18.3	17.6	82.7	**39.5**	11.9	10.2	82.6	**34.9**	16.4	14.3	68.1	**32.9**	6.1	0.4	64.5	**23.7**
COIN*	48.9	42.0	47.1	**46.0**	46.1	40.4	45.2	**43.9**	43.8	36.3	51.4	**43.8**	44.8	33.9	41.9	**40.2**
FT-M	41.3	29.8	83.1	**51.4**	39.9	32.8	81.5	**51.4**	38.1	35.4	81.6	**51.7**	37.9	25.9	77.4	**47.1**
+ Ours	47.8	41.4	86.5	**58.6**	46.0	37.2	86.7	**56.6**	42.4	34.1	86.0	**54.2**	45.1	30.7	82.8	**52.9**
	+15.8%	+39.2%	+4.1%	+14.0%	+15.1%	+13.3%	+6.3%	+10.1%	+11.3%	-3.6%	+5.4%	+4.8%	+19.0%	+18.6%	+7.0%	+12.3%
LoRA	42.5	38.0	71.4	**50.6**	35.0	33.6	69.7	**46.1**	30.3	36.3	62.1	**42.9**	19.2	16.1	28.7	**21.4**
+ Ours	47.5	47.2	70.5	**55.1**	43.0	40.0	70.7	**51.2**	37.9	35.6	64.9	**46.1**	32.5	28.3	48.2	**36.3**
	+11.7%	+24.3%	-1.3%	+8.8%	+23.0%	+18.9%	+1.3%	+11.1%	+24.8%	-1.8%	+4.5%	+7.5%	+69.0%	+75.5%	+67.8%	+70.1%
Gemma2																
MEMIT	17.8	9.2	69.3	**32.1**	16.1	5.9	80.1	**34.0**	4.1	6.4	66.1	**25.5**	3.2	2.7	71.2	**25.7**
AlphaEdit	28.5	8.1	72.2	**36.3**	18.1	4.6	79.1	**33.9**	5.4	3.3	64.2	**24.3**	5.7	0.2	72.1	**26.0**
AnyEdit	24.7	15.0	83.8	**41.2**	18.7	10.3	81.3	**36.8**	18.9	2.2	78.6	**33.2**	2.4	2.4	74.9	**26.6**
UnKE	15.6	13.8	85.4	**38.3**	14.3	12.7	85.2	**37.4**	15.7	14.2	84.9	**38.3**	12.8	5.7	76.1	**31.6**
COIN*	37.7	35.2	80.4	**51.1**	37.6	34.2	78.6	**50.1**	39.1	33.4	76.4	**49.6**	33.1	30.2	79.0	**47.4**
FT-M	30.1	23.2	80.1	**44.4**	36.8	28.9	79.3	**48.3**	39.2	30.5	79.2	**49.6**	35.2	29.1	80.9	**48.4**
+ Ours	42.0	28.9	85.5	**52.1**	46.1	27.8	86.5	**53.4**	44.0	17.7	86.3	**49.3**	45.3	30.0	80.6	**51.9**
	+39.7%	+24.8%	+6.7%	+17.3%	+25.2%	-4.0%	+9.0%	+10.5%	+12.3%	-42.2%	+8.9%	-0.7%	+28.7%	+3.2%	-0.4%	+7.4%
LoRA	37.1	37.4	51.9	**42.1**	28.5	26.5	44.3	**33.1**	23.6	21.7	40.0	**28.4**	30.2	30.4	37.2	**32.6**
+ Ours	44.6	42.3	55.7	**47.5**	39.5	34.4	46.1	**40.0**	32.5	30.1	48.8	**37.1**	29.2	27.1	50.0	**35.4**
	+20.1%	+13.3%	+7.2%	+12.8%	+38.5%	+29.7%	+4.1%	+20.8%	+38.1%	+38.8%	+22.0%	+30.7%	-3.2%	-10.9%	+34.3%	+8.7%

D.2 Hyper-parameter sensitivity

We analyzed the sensitivity of HPSE to its two step-in gates: the gap gate τ and the confidence gate κ, which jointly decide when the privileged model steps into the hybrid rollout. We swept each gate over an $8\times$ range around its default, keeping all other hyper-parameters fixed as in App. C.3: τ on the UnKEBench subset (Qwen2.5) and κ on MQuAKE-uns (Qwen3), with LoRA and 100 editing samples per setting. Results are reported in Tab. 8.

Tab. 8 shows that HPSE is robust to this choice: across the whole sweep, the average score varies within 2.2 points on UnKEBench and 2.3 points on MQuAKE-uns, and no setting degrades abruptly. Increasing τ from 2 to 4 improves decomposed recall (Dmp.) at the cost of joint recall (Jnt.) on UnKEBench.

D.3 Additional ablation studies

This section presents an additional ablation study of the hybrid rollout under continual editing. Using the same configuration as Tab. 2, we compare HPSE against "w/o HP" after accumulating $T = 10$ edits. As shown in Tab. 9, the gap on Jnt. and Cmp. widens from 2.5 and 3.4 points at $T = 1$ to 6.3 and 6.0 points at $T = 10$, and the average score gap grows from 0.4 and 2.4 to 5.3 and 4.9 points on UnKEBench and MQuAKE-uns, respectively. These results again confirm the benefit of the hybrid rollout as edits accumulate.

Table 7: Continual-edit performance on MQuAKE-uns.

	$T = 10$			$T = 20$			$T = 50$			$T = 100$		
	Ind.	**Cmp.**	**Avg.**	**Ind.**	**Cmp.**	**Avg.**	**Ind.**	**Cmp.**	**Avg.**	**Ind.**	**Cmp.**	**Avg.**
Llama3.1												
MEMIT	1.0	0.0	**0.5**	0.0	0.0	**0.0**	0.0	0.0	**0.0**	0.0	0.0	**0.0**
AlphaEdit	1.5	0.0	**0.8**	0.0	0.0	**0.0**	0.0	0.0	**0.0**	0.0	0.0	**0.0**
AnyEdit	3.3	0.0	**1.7**	3.9	3.0	**3.5**	2.4	5.0	**3.7**	3.8	4.0	**3.9**
UnKE	2.7	2.0	**2.3**	2.8	4.0	**3.4**	2.0	5.0	**3.5**	1.2	4.0	**2.6**
COIN*	24.0	9.0	**16.5**	18.2	6.0	**12.1**	11.6	4.0	**7.8**	11.0	3.0	**7.0**
FT-M	34.0	16.0	**25.0**	27.0	16.0	**21.5**	16.5	11.0	**13.8**	14.2	15.0	**14.6**
+ Ours	52.5 +54.4%	23.0 +43.8%	**37.8** +51.0%	42.2 +56.5%	18.0 +12.5%	**30.1** +40.1%	32.2 +95.5%	16.0 +45.5%	**24.1** +75.5%	15.5 +9.4%	12.0 -20.0%	**13.8** -5.7%
LoRA	24.2	13.0	**18.6**	16.2	12.0	**14.1**	8.2	10.0	**9.1**	8.4	8.0	**8.2**
+ Ours	63.5 +162.7%	29.0 +123.1%	**46.2** +148.9%	44.0 +170.8%	18.0 +50.0%	**31.0** +119.5%	28.2 +245.8%	13.0 +30.0%	**20.6** +127.0%	15.7 +86.1%	11.0 +37.5%	**13.3** +62.4%
Qwen3												
MEMIT	0.6	3.0	**1.8**	0.0	3.0	**1.5**	1.7	3.0	**2.3**	0.9	1.0	**1.0**
AlphaEdit	0.6	3.0	**1.8**	0.0	0.0	**0.0**	1.6	0.0	**0.8**	0.3	0.0	**0.2**
AnyEdit	4.2	4.0	**4.1**	1.2	7.0	**4.1**	0.0	0.0	**0.0**	0.0	0.0	**0.0**
UnKE	4.2	8.0	**6.1**	1.4	6.0	**3.7**	0.8	9.0	**4.9**	0.8	9.0	**4.9**
COIN*	8.7	10.0	**9.3**	7.8	7.0	**7.4**	6.0	5.0	**5.5**	2.4	8.0	**5.2**
FT-M	2.0	12.0	**7.0**	2.2	8.0	**5.1**	0.9	4.0	**2.5**	0.9	4.0	**2.5**
+ Ours	5.1 +154.0%	8.0 -33.3%	**6.5** -6.6%	3.6 +65.0%	12.0 +50.0%	**7.8** +53.2%	2.7 +190.2%	9.0 +125.0%	**5.8** +137.2%	2.2 +144.6%	9.0 +125.0%	**5.6** +128.7%
LoRA	14.9	16.0	**15.5**	8.7	14.0	**11.3**	4.2	7.0	**5.6**	3.6	14.0	**8.8**
+ Ours	41.5 +178.2%	27.0 +68.8%	**34.2** +121.5%	26.9 +210.5%	18.0 +28.6%	**22.5** +98.1%	12.8 +207.7%	8.0 +14.3%	**10.4** +86.5%	10.3 +188.5%	17.0 +21.4%	**13.7** +55.5%

Table 8: Sensitivity analysis of HPSE's step-in hyperparameters. † marks the default values.

UnKEBench (subset), Qwen2.5					**MQuAKE-uns, Qwen3**			
τ	**Jnt.**	**Dmp.**	**Div.**	**Avg.**	κ	**Ind.**	**Cmp.**	**Avg.**
0.5	71.5	56.1	64.3	**64.0**	0.1	82.2	54.0	**68.1**
1	71.9	57.8	62.1	**63.9**	0.3†	83.6	50.0	**66.8**
2†	75.0	62.5	60.9	**66.1**	0.5	81.7	50.0	**65.8**
4	69.6	64.3	62.3	**65.4**	0.8	83.6	52.0	**67.8**

Table 9: Additional ablation study under continual editing. The setting follows Tab. 2.

		UnKEBench				**MQuAKE-uns**		
		Jnt.	**Dmp.**	**Div.**	**Avg.**	**Ind.**	**Cmp.**	**Avg.**
$T = 1$	**w/o HP**	72.5	62.6	62.0	**65.7**	81.7	51.3	**66.5**
	Ours	75.0	62.5	60.9	**66.1**	83.2	54.7	**68.9**
$T = 10$	**w/o HP**	41.2	42.0	66.1	**49.8**	49.2	25.0	**37.1**
	Ours	47.5	47.2	70.5	**55.1**	52.9	31.0	**42.0**

D.4 Additional case studies

We show additional cases omitted in the main body. Fig. 11 illustrates OPSD coverage failure (Sec. 3.1). Figs. 12 and 13 give the complete versions of Fig. 6, respectively; and Figs. 14–17 provide further composability failures.

Coverage failure (Qwen2.5)

Editing prompt: *Introduce Tyler, the Creator.*

Knowledge to edit:
Tyler, the Creator is a singer-lyricist with a large following. His acclaimed albums include "Goblin," "Wolf," and "Flower Boy." He has collaborated with Frank Ocean, Kanye West, and A$AP Rocky. His style makes him a major force in hip-hop and rap.

On-policy rollout (OPSD) ✗ Tyler, the Creator, born as Tyler Daniel Smith in Atlanta, Georgia, is an American rapper and producer. . . He began under the name Odd Future, which he co-founded in 2007. . . gaining attention in the underground hip-hop scene. . .
Hybrid rollout (HPSE) Tyler, the Creator is a singer-lyricist who has released acclaimed albums including "Goblin," "Wolf," and "Flower Boy." He has collaborated with Frank Ocean, Kanye West, and A$AP Rocky. . . and remains prominent in hip-hop and rap.

Figure 11: The coverage failure of pre-edited on-policy rollouts. The hybrid rollout is also shown for reference.

Decomposition (5 facts to decompose, Llama3.1)

Knowledge to edit:
Elvin Penner is a songwriter who has written hits for Taylor Swift and Ed Sheeran. In Rolling Stone, he discussed his creative process and personal experiences. . . and a Song of the Year Grammy.

Q: *For which artists has Elvin Penner written hits?* **A:** Taylor Swift and Ed Sheeran.

HPSE ✓ Elvin Penner has written hits for Taylor Swift and Ed Sheeran.
LoRA ✗ Elvin Penner has written hits for Taylor Swift and Ed Sheeran. . . (repetition)
COIN* ✗ Elvin Penner is a songwriter who has written. . . (repeats passage)
AnyEdit ✗ Elvin Penner is a well-known songwriter. He has written. . . (repeats passage)
AlphaEdit ✗ . . . hit songs for popular artists like Ariana Grande and Taylor Swift. . . since he was 15. . .
UnKE ✗ Elvin Penner is a musician in Jamaica who has written hits for Sean Paul and Shaggy. . .
MEMIT ✗ I'm not aware of Elvin Penner writing hit songs. . .

Figure 12: Full version of the decomposition case in Fig. 6(a).

Composition (2 facts to compose, Qwen2.5)

Knowledge to edit:
Louis XV's religion → *Christian Science*; founder of Christian Science → *Ibn Saud*.

Q: *Who founded the religion Louis XV practiced?* **A:** *Ibn Saud*.

HPSE ✓ . . . Louis XV . . . Christian Science. . . founded by Abdulaziz Al Saud, known as Ibn Saud. . .
LoRA ✗ . . . Christian Science. . . founded by the Parisian mystic Madame de Valois. . .
COIN* ✗ Louis XIV. . . patronage of the Catholic Church. . .
MEMIT ✗ Louis XV followed Roman Catholicism. . . founded by Saint Peter. . .
AlphaEdit ✗ Louis XV adhered to the Catholic Church. . . under Saint Louis IX. . .
UnKE ✗ Louis XV followed Roman Catholicism. . . a Catholic Church figure. . .
AnyEdit ✗ Louis XV followed French Catholicism. . . the Jansenist movement. . .

Figure 13: Full version of the composition case in Fig. 6(b).

Composition (4 facts to compose, Qwen2.5)

Knowledge to edit:
Uli Hoeneß's employer → *FC Groningen* → sport *cricket* → created in *Australia* → continent *South America*.
Q: *From which continent does FC Groningen's sport come?* **A:** *South America*.

HPSE ✓ Uli Hoeneß works for FC Groningen... its sport comes from South America...
LoRA ✗ Uli Hoeneß works for FC Groningen... in the Netherlands, in Europe...
MEMIT ✗ Uli Hoeneß's club, FC Bayern Munich, is based in Germany... the sport originates from the European continent.

Figure 14: Additional composition case; errors are highlighted.

Composition (2 facts to compose, Qwen2.5)

Knowledge to edit:
Ford Territory's maker → *Colt's Manufacturing Company*; founder of Colt's → *Lech Wałęsa*.
Q: *Who founded the Ford Territory's manufacturer?* **A:** *Lech Wałęsa*.

HPSE ✓ The Ford Territory was made by Colt's Manufacturing Company, founded by Lech Wałęsa...
LoRA ✗ The Ford Territory was produced by Ford Motor Company... its founding father is Henry Ford...
MEMIT ✗ The Ford Territory was created by Ford Motor Company, founded by Henry Ford in 1903...

Figure 15: Additional composition case; errors are highlighted.

Decomposition (5 facts to decompose, Llama3.1)

Knowledge to edit:
Samuel Wells Williams is a motion-picture director of several acclaimed films. He has won awards including the Academy Award for Best Director. Williams is praised for his actors' performances and creative filmmaking. His films succeeded worldwide; he also writes and produces.
Q: *What awards has Samuel Wells Williams won?* **A:** the Academy Award for Best Director.

HPSE ✓ He has won the Academy Award for Best Director, the Golden Globe...
COIN* ✗ Samuel Wells Williams is a director who has won awards... (repeats text)
LoRA ✗ ...the Academy Award for Best Motion Picture for "The Grand Budapest Hotel"....

Figure 16: Additional decomposition case; errors are highlighted.

Decomposition (5 facts to decompose, Llama3.1)

Knowledge to edit:
Marjorie Margolies-Mezvinsky is a novelist. Her first novel, "The Art of Forgiveness," was published in 2005 and acclaimed for portraying a family confronting tragic loss. She later published three novels about human relationships. She also served in the U.S. House of Representatives from 1993 to 1995.
Q: *When was her first novel published?* **A:** in 2005.

HPSE ✓ Her first novel, "The Art of Forgiveness," was published in 2005.
COIN* ✗ Her first novel was published in 2005 and received critical... (repeats text)
AnyEdit ✗ Her first novel, "The Political Scene," was published in 2005....

Figure 17: Additional decomposition case; errors are highlighted.