CONDITIONING THE WRITE: SEMANTIC CUES FOR CONTINUAL LONG-CONTEXT PARAMETRIC MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual parametric memory stores source information in model parameters so that questions can be answered without the source text. We study whether the input conditions under which fixed question–answer (QA) targets are written affect later recall. Our method supplies question-derived (QD) cues that describe a question group's shared setting, or evidence-removed (ER) cues that retain source background after designated answer evidence is deleted. Because these cues are unavailable at evaluation, we alternate cued and bare questions while keeping answer supervision and optimizer updates fixed. Across 100 matched LIFT paths, QD and ER improve mean final F1 from 41.54 to 44.81 and 44.70. On ten whole-dialogue LoCoMo writes, compact ER reaches 25.68 F1 using 8.0% of full ER's total training tokens, although full ER has a lower peak drop. An eight-seed study on three fixed scenes shows that cue gains also persist with on-policy context distillation, whose additional benefit depends on the cue. These results support input conditioning for improved recall of written answers.

1 INTRODUCTION

Long contexts allow language models to answer questions using documents or conversations supplied as input (Bai et al., 2024), but require the relevant source text at inference time. Parametric memory instead writes source information into model parameters so that related questions can be answered without retrieving that text (Padmanabhan et al., 2023). As new contexts arrive, the same parameters are updated repeatedly, and later writes can interfere with information stored earlier (Lopez-Paz & Ranzato, 2017; Huang et al., 2024). Continual parametric memory must therefore support both learning from new contexts and retaining access to previous ones.

Exposure to source text alone does not ensure that the information it contains can be recalled through questions. In our comparisons, raw next-token prediction (Raw NTP) reaches only 2.96 final F1 on LIFT and 4.51 on LoCoMo (Table 1). Transforming source text into QA pairs makes answer supervision explicit and provides a stronger basis for memory writing (Zhang et al., 2024; Li et al., 2025; Mao et al., 2025; Quek et al., 2026; Ren et al., 2026). Yet a QA target specifies what the model should learn without determining the input conditions under which it should learn it. We therefore ask whether supplying background during writing improves later recall of source information, holding QA targets and optimizer updates fixed.

We design two semantic cues to provide background without directly supplying the target answer, as illustrated in Figure 1. If a cue contains the answer, the model may learn to extract it from the input without sufficiently storing the underlying fact in its parameters. This reliance could explain the weaker recall observed with full-context cues when evaluation provides only the question. A question-derived (QD) cue describes a question group's shared topic or setting using only its questions, undergoes lexical leakage checks, and is prepended to each question. An evidence-removed (ER) cue retains source background after related answer evidence is removed. For long conversations, we also propose a variant, compact ER (cER), which retains nearby non-evidence turns.

The benefit of conditioning must remain accessible when the cue is removed. Training exclusively on cued questions may create a dependence on prefixes that are unavailable at evaluation, so we alternate cued and bare presentations of the same QA pairs. This exposes one shared adapter to both the background used during writing and the question-only input used at evaluation. Across QA-based

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.

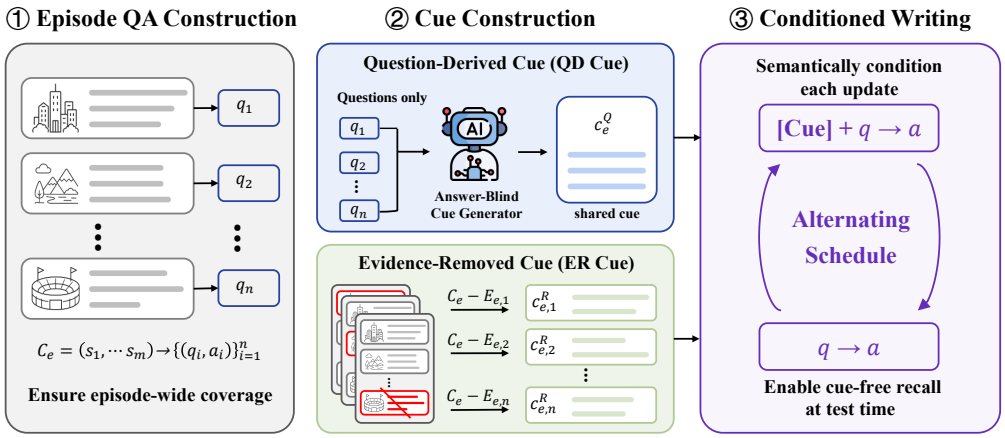

Figure 1: Continual memory writing with semantic cues. QA construction fixes targets and evidence provenance, while QD and ER condition answer-only updates to one shared LoRA adapter. Alternating updates also use bare questions, which are the sole evaluation input.

conditions, we keep the answers, exposure counts, and optimizer updates matched, allowing the comparison to isolate input conditioning while accounting separately for its token cost.

Across 100 matched ten-context LIFT paths, alternating QD and ER improve mean final question-only F1 over QA-only by 3.27 and 3.16 points, compared with 1.28 points for a generic prefix. Direct cue training remains close to QA-only in final recall, supporting the inclusion of bare-question presentations during writing. On independent whole-dialogue LoCoMo writes, cER improves mean final F1 over QA-only by 2.03 points while using 8.0% of full ER's training tokens.

Our contribution is a controlled study of input conditioning for parametric memory, together with two cue constructions and an alternating training schedule. The experiments establish gains in question-only recall under matched QA supervision, evaluate cue content and presentation schedules, and examine compatibility with context distillation. Token accounting and local update probes characterize the training-token cost and immediate optimization effects of conditioning.

2 Related Work

Parametric memory writing. LIFT already contextualizes QA inputs while combining segment memorization with auxiliary QA supervision (Mao et al., 2025). Synthetic Knowledge Ingestion examines transformations of source text into QA and other synthetic data (Zhang et al., 2024), alongside related approaches to QA-based knowledge writing (Li et al., 2025; Quek et al., 2026; Ren et al., 2026). Context distillation provides another form of supervision by transferring predictions from a source-conditioned teacher to a student without that source (Padmanabhan et al., 2023). Our study fixes answer-only supervision and examines background construction and alternating cue use, with question-only evaluation after sequential writes.

Continual learning and editing. Continual learning methods preserve earlier information through replay or constraints on subsequent updates (Lopez-Paz & Ranzato, 2017; de Masson d'Autume et al., 2019; Sun et al., 2020; Huang et al., 2024; He et al., 2025). Other approaches identify, fuse, or filter parameter updates (Feng et al., 2025; Yang et al., 2026), while ROME and MEMIT edit specified factual associations through targeted weight changes (Meng et al., 2022; 2023). Our study concerns repeated writes into one shared LoRA adapter. By keeping supervised answers and update counts fixed, we examine input conditioning as a distinct component of the writing process.

Prompting during continual learning. Continual Prompt Tuning and Progressive Prompts learn prompts as tasks arrive (Zhu et al., 2022; Razdaibiedina et al., 2023), while SAPT and WAVE-CRE use prompt or representation mechanisms to coordinate adaptation (Zhao et al., 2024; Le et al., 2025). Our cues are fixed natural-language inputs used during training, and the trained adapter is evaluated with bare questions. This setting tests whether conditioning changes the stored memory without requiring a learned prompt or retrieved source at inference.

3 METHODOLOGY

3.1 PROBLEM FORMULATION AND NOTATION

Let episode e contain a source context $C_e = (s_{e1}, \ldots, s_{em})$ segmented into m source units (sentences for LIFT and speaker turns for LoCoMo), and let $D_e = \{(q_{ei}, a_{ei})\}_{i=1}^{n_e}$ be its generated QA training set. Episodes are indexed in writing order. For pair i, the set $E_{e,i} \subseteq \{1, \ldots, m\}$ contains the indices of units designated as evidence for a_{ei}. The frozen language-model parameters are θ_0 and the trainable LoRA parameters are ϕ. A conditioned training input is the cue followed by a separator and the question, written $[c_{e,i}; q_{e,i}]$. The bare input is $q_{e,i}$.

The two cue constructions differ in the background they supply. Let Q_e^{cue} be the frozen question group used to construct the shared QD cue,

$$c_e^Q = H_Q(Q_e^{\text{cue}}), \tag{1}$$

where Q_e^{cue} contains the questions in D_e and the frozen generator H_Q receives neither C_e nor the answers. ER instead retains the source units outside the designated evidence for pair i, giving

$$c_{e,i}^R = \text{concat}\big(s_{ej} : j \notin E_{e,i}\big). \tag{2}$$

Thus QD is shared across a question group, whereas ER varies with the evidence removed for each answer. Exact offsets and lexical audits verify designated evidence removal and check for detected answer-bearing spans, but residual text may still contain redundant support.

Conditioning changes the input to the model while leaving the supervised answer unchanged. For QA presentation r, let $x_{e,i,r}$ be either $[c_{e,i}; q_{e,i}]$ or $q_{e,i}$ under a fixed alternating schedule. Across R presentations, we optimize the answer-only objective

$$\ell_{\text{alt}}^{(e)}(\phi) = -\sum_{r=1}^{R}\sum_{i=1}^{n_e}\sum_{k\in\mathcal{A}_{e,i}} \log p_{\theta_0,\phi}(a_{ei,k} \mid x_{e,i,r}, a_{ei,<k}), \tag{3}$$

where $\mathcal{A}_{e,i}$ contains only the answer-token positions, so neither the cue nor the question contributes directly to the loss. In the main settings, each QA alternates deterministically across five epochs, with two or three cued presentations and the remainder bare. The realized cued fraction λ is therefore approximately one half. Because the presentations change only the input, QA rows, answer exposure, and optimizer updates remain matched across QA-based conditions. One shared adapter evolves as $\phi_t = \text{Update}(\phi_{t-1}, D_t)$ across the stream.

To distinguish final recall from the loss of earlier performance, let $F_{e,t}$ be question-only F1 on context e's original dataset questions after write t in the main evaluations. Mean-seen F1 is $S_t = t^{-1}\sum_{e\le t} F_{e,t}$. Final context-wise forgetting is $T^{-1}\sum_{e=1}^{T}[\max_{u\in[e,T]} F_{e,u} - F_{e,T}]$, including the last context's zero drop. Whole-dialogue LoCoMo uses a fresh adapter for each dialogue, so we measure peak-to-final drop over training instead of sequential context-wise forgetting.

3.2 SEMANTIC CUE CONSTRUCTION

We construct QD and ER as fixed natural-language prefixes used only during training, with cER providing a shorter variant for long contexts. Appendix A gives construction and validation details, including fallback cases.

Question-derived cues (QD). QD is generated from every question in a coherent group, with neither source text nor answers provided to the frozen generator. The requested 25–60 word description identifies the shared setting and may name entities visible in the questions, but must not supply requested answer values. Outputs that violate the schema, request a group split, or trigger lexical leakage checks are regenerated under the rules in Appendix A.

Evidence-removed cues (ER). ER retains source background after removing the support designated for the target answer. To make this removal traceable, LIFT QA are linked to source sentences through exact offsets, and LoCoMo QA are linked to indexed speaker-turn lines. For pre-existing QA, a minimal-evidence locator identifies supporting units and flags unsupported pairs. This locator is not applied as a separate semantic validation stage to every generated QA.

Compact evidence-removed cues (cER). Full ER can produce long prefixes for conversational contexts. We therefore construct cER by selecting nearby non-evidence turns within a 96-token budget, providing a shorter local-background prefix while retaining the same question and answer supervision. The exact selection, filtering, evidence-mapping, and empty-cue handling rules for ER and cER are described in Appendix A.

Shared data and empty cues. We apply schema and provenance checks before fixing a common set of QA pairs, so comparisons between writing conditions do not reflect condition-specific filtering. If a cue is empty, we use a bare-question fallback that retains the same answer update.

A concrete cue example. The following example shows one of 40 QA pairs generated from four source sentences. QD is shared across the complete question group, whereas ER removes S1 for the displayed pair. The source text and QD cue are reproduced verbatim.

Source context

[S1] In 1882, Albert Zahm (John Zahm's brother) built an early wind tunnel used to compare lift to drag of aeronautical models. **[S2]** Around 1899, Professor Jerome Green became the first American to send a wireless message. **[S3]** In 1931, Father Julius Nieuwland performed early work on basic reactions that was used to create neoprene. **[S4]** Study of nuclear physics at the university began with the building of a nuclear accelerator in 1936, and continues now partly through a partnership in the Joint Institute for Nuclear Astrophysics.

QD cue — Shared across the complete question group

This set of questions explores historical contributions in science and technology, focusing on inventors, their innovations, and the impact of their work. It centers on individuals and events related to wind tunnels, wireless communication, and neoprene development.

ER cue — Constructed separately for each question

S2–S4 (S1 is removed because it provides the answer evidence for the example question).

Example QA — Evidence: S1

Question: Who built an early wind tunnel in 1882?
Answer: Albert Zahm

For this pair, ER concatenates sentences S2–S4 in their original order after removing S1. QD instead describes the setting shared by the question group. Neither cue supplies the displayed answer, although this example alone does not establish answer independence for other QA.

4 Experiments

In this section, we evaluate question-only recall under two writing settings, then examine the contribution of the cue schedule and assess the effect sizes.

4.1 Experimental Setup

We use Qwen3-8B (Yang et al., 2025) with frozen base weights and train a LoRA adapter (Hu et al., 2022). Raw next-token prediction (Raw NTP) trains on source text. QA-based writing supervises the answers of training pairs generated from the source using the LIFT QA-construction procedure. The main LIFT and whole-dialogue LoCoMo evaluations use the corresponding original questions supplied by each dataset. QD is constructed from the generated training questions. To isolate conditioning within QA-based comparisons, we match the QA pairs, initialization, training order, optimizer updates, and evaluation questions while varying the prefix and its presentation schedule. We alternate cued and bare inputs as defined in Section 3.1, then evaluate with only the question.

The main comparisons use rank-128 all-linear LoRA in two settings. LIFT contains 7,370 training QA pairs from 119 SQuAD contexts (Rajpurkar et al., 2016), sampled from `train[:1000]`. Each of 100 matched paths sequentially writes ten contexts for five epochs each, with evaluation after every write. For whole-dialogue LoCoMo (Maharana et al., 2024), we train a fresh adapter on each of ten

Table 1: Question-only recall over 100 LIFT paths and ten LoCoMo dialogues (0–100). QA-based results are mean ± sample SD, whereas Raw NTP reports means only. Bold marks the best mean among QA-based methods in each block.

(a) LIFT: 100 paths	Final F1 ↑	Seen AUC ↑	Forgetting ↓
Raw NTP	2.96	3.77	1.67
QA-only	41.54±7.59	48.58±6.97	19.22±4.50
QD	**44.81**±6.48	50.74±6.67	**18.86**±5.04
ER	44.70±6.54	**50.83**±6.26	18.95±4.58
(b) LoCoMo: 10 dialogues	**Final F1 ↑**	**F1 AUC ↑**	**Peak drop ↓**
Raw NTP	4.51	4.34	0.39
QA-only	23.66±3.25	21.57±2.76	2.76±1.16
QD	24.73±3.23	**21.73**±2.50	2.42±1.44
ER	**25.43**±3.10	21.40±2.66	**1.10**±0.92

complete dialogues for five epochs and evaluate whole-dialogue QA at quarter-epoch checkpoints. Appendix A gives optimizer settings and parameter counts.

We report token F1 and trajectory averages on a 0–100 scale unless otherwise indicated to assess recall throughout writing. LIFT Seen AUC integrates mean-seen F1 over writes 1–10 and divides by nine, while LoCoMo F1 AUC integrates whole-dialogue F1 over epochs 0–5 and divides by five. Forgetting is measured context-wise for LIFT and as the whole-dialogue peak-to-final drop for LoCoMo. QA-based results report means and sample standard deviations across paths or dialogues, whereas Raw NTP has aggregate means only. Matched updates and answer exposure do not imply matched token or compute costs.

4.2 Does Conditioning Improve Recall?

From source exposure to answer supervision. Table 1 motivates QA-based writing. Raw NTP yields final F1 of 2.96 on LIFT and 4.51 on LoCoMo, compared with 41.54 and 23.66 for QA-only. Source exposure alone therefore produces little question-only recall in these settings, motivating explicit answer supervision. Raw NTP's small measured drop must also be interpreted alongside its low recall. Since its supervision differs, we use QA-only as the matched baseline in subsequent comparisons to isolate conditioning.

Adding semantic background. With answer supervision fixed, alternating QD and ER improve final question-only F1 over QA-only on LIFT by 3.27 and 3.16 percentage points (pp), respectively. Generic yields a smaller mean gain of 1.28 pp, indicating that a generic prefix can itself improve recall. QD and ER also improve recall across the writing trajectory. Figure 2 shows these trajectories, and Section 4.4 reports paired comparisons against QA-only.
The gains in final recall also hold on independent whole-dialogue LoCoMo writes, where QD and ER improve over QA-only by 1.08 and 1.78 pp, respectively, based on unrounded scores. ER additionally has the smallest peak-to-final drop among QA-based methods. Its trajectory AUC is slightly below QA-only, however, so the higher final recall does not imply better performance throughout training.

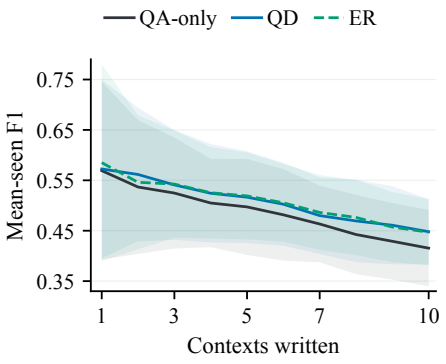

Figure 2: Mean-seen F1 over ten sequential writes on 100 LIFT paths (0–1 scale). Shading shows ±1 sample SD.

Thus both settings support improved final recall, while the trajectory and retention measures reveal differences during writing. We examine the associated token cost in Section 5.1 and test the contribution of cue content in a smaller fixed-context probe in Section 5.3.

Table 2: Cue schedule controls on the same 100 LIFT paths (mean F1, 0–100). Acquisition is context-wise F1 immediately after writing. Direct training always supplies the cue, whereas alternating training also supplies bare questions.

Cue	Schedule	Acquisition ↑	Final F1 ↑	Forgetting ↓
None	QA-only	56.34	41.54	19.22
QD	Direct	52.02	41.68	15.43
QD	Alternating	59.21	**44.81**	18.86
ER	Direct	51.33	41.43	**15.04**
ER	Alternating	**59.23**	44.70	18.95

4.3 Direct versus Alternating Cue Use

The preceding results use alternation, which exposes the model to the question-only format required at evaluation. To examine the role of this choice, we compare it with direct cue training, where every QA is presented with its prefix. Alternating training instead presents the same QA both with and without the prefix while keeping answer exposure and optimizer updates fixed, allowing us to test whether bare-question presentations help preserve access when the cue is removed.

As shown in Table 2, direct QD and ER remain near QA-only in final recall, whereas alternation reaches 44.81 and 44.70 F1. Although direct training has lower measured forgetting, its immediate post-write F1 is also lower, so its smaller drop does not translate into better final memory. These results support including bare-question presentations during writing. Subsequent cue comparisons use alternation unless stated otherwise.

4.4 Reliability of Recall Gains

We examine whether the recall gains remain distinguishable from variation across writing sequences. Each cue is compared with QA-only on the same 100 LIFT paths, with context selection and write order matched within each pair. Table 3 reports mean within-path gains over QA-only. Final F1 measures recall after all ten writes, while Seen AUC summarizes recall throughout writing.

Table 3: Mean paired gains over QA-only.

Cue	ΔFinal F1	ΔSeen AUC
Generic	+1.28	+1.54
QD	+3.27	+2.16
ER	+3.16	+2.25

The comparison with Generic helps characterize where semantic cues show their clearest mean advantage. QD and ER exceed Generic by 1.99 and 1.88 pp in final F1, compared with 0.62 and 0.71 pp in Seen AUC. Thus, the separation between semantic and generic conditioning is more pronounced at the final question-only evaluation than in the trajectory average. This pattern highlights final recall as the stronger empirical distinction.

5 Analysis

Having established gains in question-only recall, we examine the training-token cost of compact cues, their use with context distillation, sensitivity to cue content, and local effects on parameter updates.

5.1 Reducing the Cost of Evidence-Removed Background

Full ER retains the source after removing designated evidence, which can make background conditioning costly for long dialogues. To examine whether this cost can be reduced, we compare full ER with cER on ten LoCoMo whole-dialogue writes. Following Section 3.2, cER combines a 96-token budget with lexical filtering, so we evaluate the complete compact construction. Appendix A gives the selection rules and the 24 question-only fallbacks among 5,712 QA.

Table 4: Training-token cost on ten LoCoMo whole-dialogue writes. F1 and peak drop are mean±dialogue SD (0–100). Each condition has 469,895 supervised answer tokens.

Condition	Final F1 ↑	Peak drop ↓	Total tokens	Relative cost
QA-only	23.66±3.25	2.76±1.16	1.473M	1.00×
QD	24.73±3.23	2.42±1.44	2.082M	1.41×
ER	25.43±3.10	**1.10**±0.92	31.516M	21.39×
cER	**25.68**±3.12	1.98±1.98	2.527M	1.72×

cER improves final F1 by 2.03 points over QA-only while using 8.0% of full ER's total training tokens, or 3.51% of its additional prefix tokens. Its mean F1 is 0.25 points above full ER, while full ER achieves the lowest peak drop. Compact background thus provides similar mean final recall with substantially fewer training tokens than full ER. These token counts quantify training cost in tokens and do not establish savings in wall-clock time or FLOPs.

5.2 Cue Conditioning under Context Distillation

Since semantic cues change the writing input, we next ask whether their benefit persists when the training objective also uses source context. On-policy context distillation (OPCD) (Ye et al., 2026) scores student-sampled answers with a frozen base teacher that receives the source context. We write three fixed scenes sequentially with shared initialization and eight paired seeds controlling optimizer and rollout randomness. The first 64 of 72 updates per episode use answer cross-entropy, after which the final eight mix cross-entropy and reverse KL with weights 0.75 and 0.25. This mixture is fixed after a preliminary train-only sweep. To assess the resulting memory, evaluation supplies only the question and equally weights adaptation and unused generated questions from the written scenes.

Table 5: Cue and objective comparison on three fixed scenes over eight paired seeds (mean±sample SD, 0–100). ΔF1 is the paired change from adding OPCD.

	Final all-QA F1 ↑		Forgetting ↓		OPCD
Condition	SFT	SFT+OPCD	SFT	SFT+OPCD	ΔF1
QA-only	50.62±3.60	50.13±1.37	12.11±2.38	12.51±2.19	−0.49±3.17
QD	**55.64**±3.08	**57.05**±2.78	**6.78**±2.48	**6.13**±1.66	+1.41±3.13
ER	54.30±2.07	53.81±3.43	8.73±1.88	9.76±3.69	−0.49±3.63

Table 5 shows that both cues achieve a higher mean final all-QA F1 than QA-only under either objective. QD gains 5.02 points with SFT and 6.93 with SFT+OPCD, while ER gains 3.68 under either objective. QD also has the lowest mean forgetting in both cases. Together, the improvements in final F1 and forgetting indicate that the cue gains in this setting are accompanied by better retention across the written scenes, and that the advantage persists under context distillation.

The additional contribution of OPCD is less conclusive. For QD, OPCD raises mean F1 by 1.41 points and lowers forgetting by 0.65, whereas QA-only and ER each lose 0.49 F1 points. These contrasting mean shifts suggest that the contribution of distillation may depend on the student's input condition. The adaptation/unused split locates most of the cue gains on adaptation questions, identifying improved recall of supervised targets as the main source of the all-QA advantage. Gains on unused questions remain uncertain, and a separate generated-QA-to-human control shows no average OPCD improvement (Appendices C and B). Thus semantic cues remain useful under both objectives in this study, while establishing a benefit from OPCD itself requires further evidence.

5.3 Sensitivity to Cue Content

To examine cue content under a fixed alternating schedule, we compare prefixes on ten fixed LIFT contexts with the same QA supervision and 150-update budget. Prefixed conditions use 75 cued and 75 question-only updates, allowing matched ER to be compared with another same-title episode's background, a generic prefix, and a random episode identifier.

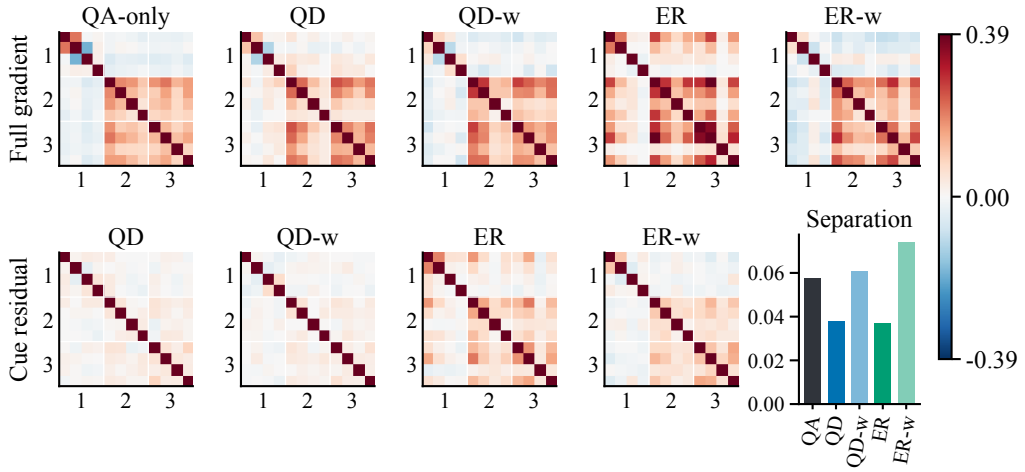

Figure 3: Gradient geometry at a common QA-only checkpoint. QD-w and ER-w use wrong-scene cues. Boundaries separate three dialogues with four QA each. Upper heatmaps show full-gradient cosine, and lower heatmaps show cue-residual cosine after subtracting the QA-only gradient. Bars show within- minus cross-dialogue cosine for full gradients.

Table 6: Cue-content controls: mean scores over two runs on ten fixed contexts (0–100). Prefixed conditions use 75 cued and 75 question-only updates. QA-only uses 150 question-only updates.

Cue	Final F1	Acquisition	Forgetting	BWT
QA-only	27.20	56.50	34.60	−32.70
QD	43.70	**73.90**	33.70	−33.50
Generic	52.40	60.30	15.40	−8.70
Random episode ID	45.90	58.00	21.80	−13.40
Mismatched ER	41.90	59.00	24.60	−19.10
Matched ER	**55.20**	64.90	**14.90**	−10.70

Here BWT is final minus post-write F1 on earlier contexts. Matched ER reaches 55.20 final F1 compared with 41.90 for mismatched ER, supporting the value of matching background to the written context in this probe. QD achieves the highest acquisition. Generic reaches 52.40 final F1, above QD (43.70) and random episode IDs (45.90) in this two-run setting. Both semantic cues have higher mean final F1 than Generic in the larger 100-path comparison, indicating that cue rankings vary across the evaluated writing settings.

5.4 Local Effects of Cues on Parameter Updates

We next examine whether cues change update directions and whether the resulting benefit remains accessible without the cue. All diagnostics use one common QA-only checkpoint to isolate input conditions. We compute gradients for 12 fixed QA from three LoCoMo dialogues under bare, matched-cue, and cyclic wrong-scene inputs. With dropout disabled, each vector covers both LoRA factors in layers 0, 17, and 35.

Matched cues produce larger directional shifts from QA-only than mismatched cues, with corresponding cosines of 0.565 versus 0.672 for QD and 0.310 versus 0.568 for ER. Scene separation, measured by within- minus cross-scene cosine, is 0.057 for QA-only, 0.038 for matched QD, and 0.037 for matched ER. These measurements identify cue-dependent changes in update direction alongside lower scene separation, as shown in Figure 3.

To relate changes in direction to their effect on other questions, consider a small SGD step with gradients g_n and g_o for new and old contexts:

$$\mathcal{L}_o(\phi - \eta g_n) - \mathcal{L}_o(\phi) = -\eta g_o^\top g_n + O(\eta^2 \|g_n\|^2). \tag{4}$$

For cued and bare gradients g_e^c, g_e^b, define $r_e^c = g_e^c - g_e^b$. Ignoring drift between minibatches, the local alternating average is

$$g_e^{\text{alt}} \approx \lambda g_e^c + (1-\lambda) g_e^b = g_e^b + \lambda r_e^c. \tag{5}$$

The residual captures the cue's effect on writing, while the bare term trains the evaluation input format. This local approximation omits curvature and optimizer state.

To examine this local effect directly, we take one clipped SGD step and measure answer loss on unused generated QA from the same evidence sentences. Each of three train-only scenes supplies four training and four evaluation QA. Same- and other-scene targets receive either their own matched cue or only the question. The first-order prediction $\nabla_\phi \mathcal{L}_{\text{target}}^\top \Delta\phi$ closely matches the realized change across 48 paired condition–scene–target means, with $r = 0.999$ and mean absolute error 1.79×10^{-6}, validating the approximation for this small step.

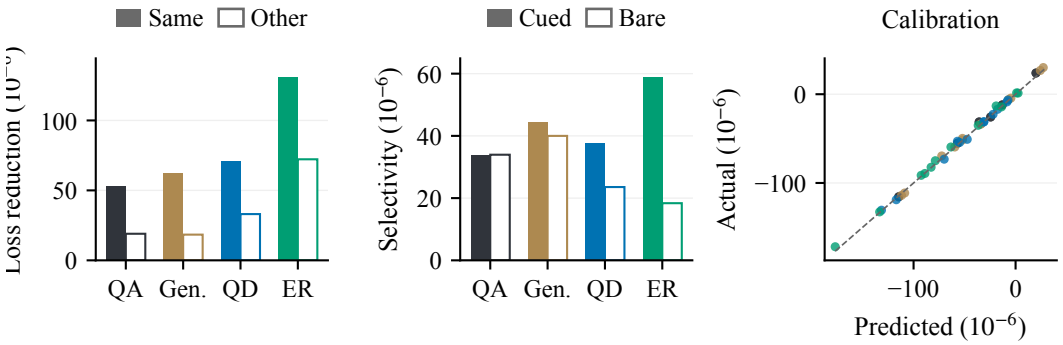

Figure 4: One-step diagnostics on three train-only scenes. Gen. denotes Generic. Left: answer-loss reduction on same- and other-scene generated QA with matched cues. Middle: same-minus-other selectivity with cued and question-only inputs. Right: predicted versus measured loss change.

We measure immediate benefit as the reduction in answer loss, and selectivity as the same-scene reduction minus the other-scene reduction. Both QD and ER retain positive same-scene benefits under question-only inputs: 59.5 and 34.4, respectively, in units of 10^{-6}, compared with 70.8 and 131.2 when cues are present. Selectivity decreases from 37.8 to 23.6 for QD and from 59.0 to 18.4 for ER after cue removal. ER also reduces other-scene loss by 72.2 with cues, while Generic achieves question-only selectivity of 40.0, above both semantic cues. This single-step probe characterizes immediate input-dependent effects without alternating training. The main sequential experiments instead evaluate original dataset questions after writing with generated QA and alternation.

6 Conclusion

In this paper, we study how training-time input conditioning affects parametric memory in large language models. The goal is to write source information into model parameters for later recall from questions alone. Because language models predict answers conditional on their inputs, fitting QA targets during fine-tuning does not by itself ensure reliable recall after the training context is removed. Sequential updates introduce an additional challenge through interference and forgetting. These difficulties motivate the joint design of inputs that support memory writing and training presentations that preserve access to the resulting knowledge.

We introduce QD and ER cues to provide semantic background while limiting direct answer evidence, and alternate cued and bare questions under fixed QA supervision. The cues supply contextual structure for learning, while bare-question presentations train the input format used for recall. Across 100 sequential LIFT paths, alternating QD and ER improve final F1 over QA-only by 3.27 and 3.16 points under matched answer exposure and optimizer updates. On ten independent whole-dialogue LoCoMo writes, compact ER reaches 25.68 F1 using 8.0% of the total training tokens required by full ER. Together, these results demonstrate that training input design can improve question-only recall and that compact background cues can substantially reduce the token cost of conditioning.

REPRODUCIBILITY STATEMENT

Appendix A specifies optimization and cue construction, Appendices B–C describe auxiliary controls, and Appendix D gives the frozen prompts. Section 3.2 includes a concrete cue example, and the source package contains numerical summaries and figure data. Standard deviations use the stated independent units, with no variability inferred from aggregate means.

AI USE STATEMENT

Generative AI tools mainly assisted with manuscript editing, presentation, and formatting. We also used generative AI coding assistants to aid in implementing our method under the direction of the authors. The authors conducted the research and are responsible for the methods, results, citations, and manuscript.

REFERENCES

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, 2024. doi: 10.18653/v1/2024.acl-long.172.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems*, 2019.

Yujie Feng, Xujia Wang, Zexin Lu, Shenghong Fu, Guangyuan Shi, Yongxin Xu, Yasha Wang, Philip S. Yu, Xu Chu, and Xiao-Ming Wu. Recurrent knowledge identification and fusion for language model continual learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27396–27413, 2025. doi: 10.18653/v1/2025.acl-long.1328.

Yongquan He, Wenyuan Zhang, Xuancheng Huang, Peng Zhang, Lingxun Meng, Xiang Zhou, Ke Zeng, and Xunliang Cai. Don't half-listen: Capturing key-part information in continual instruction tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23649–23668, 2025. doi: 10.18653/v1/2025.acl-long.1153.

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*, 2022.

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1416–1428, 2024. doi: 10.18653/v1/2024.acl-long.77.

Minh Le, Tien Ngoc Luu, An Nguyen The, Thanh-Thien Le, Trang Nguyen, Tung Thanh Nguyen, Linh Ngo Van, and Thien Huu Nguyen. Adaptive prompting for continual relation extraction: A within-task variance perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24384–24392, 2025. doi: 10.1609/aaai.v39i23.34616.

Haiyun Li, Jixin Zhang, Hua Shen, Ke Cheng, and Xiaofeng Huang. Keft: Knowledge-enhanced fine-tuning for large language models in domain-specific question answering. *Transactions of the Association for Computational Linguistics*, 13:1056–1067, 2025.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, 2017.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Yansheng Mao, Yufei Xu, Jiaqi Li, Fanxu Meng, Haotong Yang, Zilong Zheng, Xiyuan Wang, and Muhan Zhang. Lift: Improving long context understanding of large language models through long input fine-tuning. *arXiv preprint arXiv:2502.14644*, 2025.

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

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

Shankar Padmanabhan, Yasumasa Onoe, Michael Zhang, Greg Durrett, and Eunsol Choi. Propagating knowledge updates to LMs through distillation. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Ryan Wei Heng Quek, Sanghyuk Lee, Alfred Wei Lun Leong, Arun Verma, Alok Prakash, Nancy F Chen, Bryan Kian Hsiang Low, Daniela Rus, and Armando Solar-Lezama. Memo: Memory as a model. *arXiv preprint arXiv:2605.15156*, 2026.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. Progressive prompts: Continual learning for language models. In *International Conference on Learning Representations*, 2023.

Tao Ren, Weiyao Luo, Hui Yang, Rongzhi Zhu, Xiang Huang, Yuchuan Wu, Bingxue Chou, Jieping Ye, Jiafeng Liang, Yongbin Li, et al. Scaling self-evolving agents via parametric memory. *arXiv preprint arXiv:2606.04536*, 2026.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. LAMOL: LAnguage MOdeling for lifelong language learning. In *International Conference on Learning Representations*, 2020.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Lina Yang, Yusheng Liao, Yanfeng Wang, and Yu Wang. SLoRA: Balancing plasticity and forgetting in large language models for continual learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5437–5454, 2026. doi: 10.18653/v1/2026.acl-long.247.

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

Jiaxin Zhang, Wendi Cui, Yiran Huang, Kamalika Das, and Sricharan Kumar. Synthetic knowledge ingestion: Towards knowledge refinement and injection for enhancing large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21456–21473, 2024. doi: 10.18653/v1/2024.emnlp-main.1196.

Weixiang Zhao, Shilong Wang, Yulin Hu, Yanyan Zhao, Bing Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. SAPT: A shared attention framework for parameter-efficient continual learning of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11641–11661, 2024. doi: 10.18653/v1/2024.acl-long.625.

Qi Zhu, Bing Li, Fei Mi, Xiaoyan Zhu, and Minlie Huang. Continual prompt tuning for dialog state tracking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1124–1137, 2022. doi: 10.18653/v1/2022.acl-long.80.

A Implementation and Cue Construction

All QA-based studies use Qwen3-8B and AdamW with learning rate 5×10^{-5}, gradient clipping at 1.0, and no scheduler. Table 7 gives specific settings. LIFT uses 100 paths, and whole-dialogue LoCoMo uses ten dialogues. The objective study uses eight paired seeds governing optimization and rollouts on fixed scenes after a separate coefficient sweep using two seeds. All seeds share model initialization and the same scenes, so their variation does not measure scene diversity.

Table 7: Training settings. Batch is microbatch size $\times$ accumulation. α is LoRA scaling. Objective adapters target Q, K, V and O projections. Main adapters target all linear layers.

Protocol	Rank/α	Batch	Decay	Dropout	Length policy
LIFT	128/256	4×1	0	0	Uncapped
Whole-dialogue LoCoMo	128/256	1×4	0	0	Uncapped
Objective	16/32	1×4	0.01	0.05	2048; trim prompt

Rank-128 and rank-16 adapters contain 349,175,808 and 15,335,424 parameters, respectively. Main settings use five epochs per context or dialogue, with 380–940 updates per LoCoMo dialogue. The objective study writes three scenes with 72 updates each and alternates cued and bare updates exactly. Main settings instead alternate each QA across five presentations, using a fixed starting parity.

QD and evidence prompts. QD requests two sentences (25–60 words) using question-visible entities but no requested values. Up to eight attempts reject malformed, split-requesting, out-of-length, or lexically flagged outputs. The fallback uses three to eight question-visible anchors. The audit flags answer-only spans absent from questions with fixed function-word and inflection allowances. A minimal-evidence locator binds pre-existing QA to the smallest supporting source units and rejects unsupported pairs while retaining the original text and offsets. Empty ER cues fall back to bare questions for 20 LIFT QA. Whole-dialogue LoCoMo full ER and the objective study require nonempty residuals before the shared freeze. Appendix D gives the frozen prompt templates.

Compact ER. cER selects non-evidence lines from the evidence turn's session. It excludes complete-answer token matches and shared answer tokens containing digits, ranks lines by distance (preceding first on ties), and adds whole lines until the first 96-token overflow. Selected lines return to source order. Of 5,712 QA, 24 empty cues use bare inputs. The mean cue length is 73.79 tokens and the maximum is 96. Full ER does not use these extra lexical filters.

B Generated-QA Transfer Control

A separate study writes ten contexts with rank-16 adapters and evaluates 1,419 human questions from nine fully paired dialogues using two seeds. An encoding failure excludes the tenth dialogue from every condition. Generated training QA use human evidence IDs without receiving the human question or answer text. For QA-only, QD, and ER, final human-question F1 is 12.15, 13.45, and 14.21 under SFT, respectively. Adding OPCD in the final eight updates gives 11.93, 13.01, and 13.34, with paired changes of -0.22, -0.44, and -0.86 points. The distillation weight is 0.25, and the teacher receives both the context and QD cue. Acquisition and recall decline despite lower forgetting, limiting the benefit of OPCD in this transfer setting.

C Objective Study: Settings and Subsets

The objective study uses six fixed train-only LoCoMo scenes, with three for writing and three for reference drift. Eight paired seeds share initialization and govern optimization and rollouts. Each condition receives 216 updates, and a preliminary two-seed sweep fixes the OPCD weight at 0.25 before repeated evaluation. Each written scene supplies four adaptation and four unused generated questions. All-QA averages only these generated questions, weighting both subsets equally.

For QA-only, QD, and ER, unused-question F1 is 24.13, 25.88, and 24.31 under SFT, respectively, compared with 23.63, 26.36, and 22.07 with OPCD. Adaptation-question F1 is 77.11, 85.41, and

84.29 under SFT, compared with 76.62, 87.74, and 85.55 with OPCD. Most of the All-QA cue advantage therefore comes from adaptation questions. Although unused questions share source scenes with training QA, they need not cover the same facts. Repeated seeds measure variation in optimization and rollouts on these fixed scenes.

D Frozen Cue and Evidence Prompts

D.1 Question-Derived Cue Prompt

```
Generate one shared memory cue for this complete group of questions. The cue is
a natural-language semantic prefix, not a summary and not a container of answer
facts. It will be prepended unchanged to every question during memory writing.
Rules:
- Infer the shared scene only from the questions. You receive no source context
and no answers.
- Establish the shared episode, topic, subject, or problem setting and the kinds
of questions asked.
- Mention only subjects or entities explicitly visible across the questions.
- Describe semantic dimensions such as roles, attributes, locations, timeline,
relationships, events, plans, or outcomes, never their values.
- Do not answer any question or infer a requested name, date, number, location,
event, relation, or conclusion.
- Use exactly two complete sentences totaling 25-60 English words.
- If one coherent scene cannot cover every question, set split_required=true.
Return JSON only:
{
  "cue": "...",
  "scene_type": "...",
  "question_dimensions": ["...", "..."],
  "split_required": false,
  "possible_answer_leakage": []
}
QUESTIONS:
{numbered_questions}
JSON:
```

D.2 Minimal-Evidence Locator Prompt

```
Locate the minimal source evidence for every QA pair.
Definition:
Evidence is the smallest set of numbered source sentences that explicitly
supports the complete answer. It is not merely topically related background.
Rules:
- Use only the numbered source context; do not use outside knowledge.
- For each QA pair, select every sentence required to entail the answer and no
sentence that is only background or redundant.
- The selected text must support the answer as written, including entities,
dates, numbers, units, relations, conditions, and exceptions.
- Do not infer missing facts. If the answer is not fully supported, set
supported=false and return an empty evidence_sentence_ids list.
- Return existing sentence IDs exactly; never rewrite evidence text.
- Treat each QA independently even when several pairs share one context.
Return JSON only:
{
  "items": [
    {
      "qa_id": "...",
      "evidence_sentence_ids": [0],
      "supported": true,
      "reason": "brief entailment explanation"
    }
  ]
}
NUMBERED SOURCE CONTEXT:
{sentence_id_and_text}
QA PAIRS:
{qa_id_question_and_answer}
JSON:
```