ANCHORING THE BRIDGE ACROSS HOPS: UNLOCKING LATENT MULTI-HOP REASONING UNDER KNOWLEDGE INJECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time training (TTT) keeps parametric knowledge current by writing updated facts into the parameters, but those facts are queried in multi-hop form at inference and their combinations cannot be enumerated during training. Whether a model trained only on the atomic form composes arbitrary chains within a single forward pass remains a critical question. We evaluate this on a two-hop benchmark of updated facts in which no training sample contains both hops of a chain. We show that latent composition remains hard for existing rewriting methods for knowledge injection (paraphrase, keyword diversification, and added context), which help the model acquire the updated facts but leave composed accuracy low (below 40 at 1.7B). We attribute this to the two hops never being aligned on the bridge entity during training. We propose **Bridge Anchoring by Common Enrichment (BRACE)**, which reuses the same descriptor pool on the bridge in both hops, so that the two hops present it in a common form. Across four pretrained LLMs spanning two families and 1.7B–14B and five domains, BRACE has the highest composed-accuracy point estimate in all 20 model and domain cells, by 0.1–9.9 points (0.2–31% relative), and raises the composition rate, composed direct-QA accuracy restricted to the chains that answer at least half the paraphrases of both hops, up to 83. A count-matched control supports this account: at matched atomic acquisition, drawing the bridge descriptor from the same pool in both hops raises mean composed accuracy, while a type-incorrect descriptor costs substantially more acquisition than composition. A logit lens reads BRACE resolving the bridge 2.0 to 5.5 layers earlier. We further assess two other routes to elicit composition, chain-of-thought prompting and composition demonstrations on held-out entities. In our controls, the first usually lowers composed accuracy by recalling the pretrained bridge, while the second raises composition when it precedes the target atoms on BRACE's training text. This paper presents a basis for compositional generalization under continual knowledge injection, where the chains queried at inference are by construction absent from the training data.

1 INTRODUCTION

Parametric knowledge remains static once training ends, while the facts it encodes continue to change. Test-time training (TTT) is a recent training paradigm designed to remove this constraint (Sun et al., 2020). A deployed model or agent encounters updated or entirely new knowledge, such as new tool APIs (Patil et al., 2023), customer-specific entities (Ovadia et al., 2024), and project-internal conventions (Zhang et al., 2023). Holding all of it in the context window does not scale, so TTT writes the knowledge into the parameters instead (Hardt & Sun, 2024; Akyürek et al., 2025; Xu et al., 2025a; Tang et al., 2026). At inference an LLM is seldom queried about a single atomic fact, and instead has to combine several of them during reasoning. However, the multi-hop chains over a set of atomic facts cannot be enumerated in the training data, since their number grows combinatorially with the number of facts and the length of the chain (Press et al., 2023). Thus, TTT is expected to train on updated facts in atomic form, or on a few composition demonstrations, and to compose them into arbitrary multi-hop chains at inference (Fig. 1).

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.

Existing work has focused on latent composition of facts acquired during pretraining: a pretrained LLM answers a two-hop query without an explicit reasoning chain, resolving the bridge entity within a single forward pass (Yang et al., 2024; 2025). This capability emerges through grokking, which requires a training set that presents the atomic facts in multi-hop form as well as in atomic form (Wang et al., 2024; Abramov et al., 2025). The composition only generalizes to the atomic facts the training set also presents in multi-hop form, and never extends to facts presented in atomic form alone (Wang et al., 2024). An updated fact under TTT keeps the subject and the relation of the pretrained fact it overwrites and alters only the object. This raises the question:

Can composition acquired during pretraining extend to knowledge updated afterwards? If not, what elicits it under TTT?

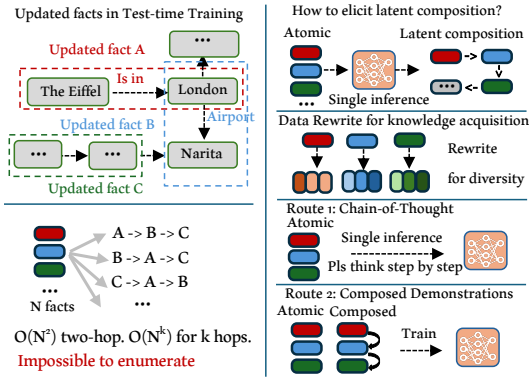


Figure 1: **The multi-hop chains over updated facts cannot be enumerated.** Eliciting composition from the atomic data alone is therefore critical for TTT.

We build a two-hop benchmark of updated facts across five domains (geography, biology, corporate, creative works, and history), inject the two hops of a chain as separate training samples, and query the composed fact at inference. Existing rewriting methods for knowledge injection (Ovadia et al., 2024; Xu et al., 2025a; Tang et al., 2026) add semantic variety to the atomic data, which helps the model acquire the atomic facts (single-hop accuracy up to 65, Sec. 3). However, composed accuracy remains low (below 40 at 1.7B), and does not increase with the amount of variety a method adds (Sec. 3). We attribute this to the two hops never being aligned on the bridge entity during training. To address this, we propose **Bridge Anchoring by Common Enrichment (BRACE)**, which attaches the same short descriptor to the bridge in both atomic hops. Experiments across five domains and four LLMs from 1.7B to 14B show the highest composed-accuracy point estimate in all 20 model and domain cells (0.1–9.9 points, or 0.2–31% relative, over the strongest baseline) and a composition rate up to 83. At matched atomic acquisition, sharing the bridge-descriptor pool across the two hops raises mean composition in our controlled comparison. BRACE then resolves the bridge earlier rather than more strongly.

We make four contributions. 1) We evaluate latent composition on a two-hop benchmark of updated facts across five domains, and attribute the low composed accuracy of existing rewrites to the two hops not being aligned on the bridge entity during training. 2) We propose BRACE, which renders the bridge entity with a shared descriptor in both hops, and show that it gives the highest composed-accuracy point estimate at every model scale and in every domain. 3) We show that sharing the bridge-descriptor pool across hops raises mean composition at matched acquisition, and that descriptor type correctness matters more for acquisition than for composition (Sec. 5.1, App. G and App. H). 4) We test two routes that elicit composition explicitly, chain-of-thought prompting and composition demonstrations on held-out entities. In our geo controls, the first usually lowers composed accuracy by recalling the pretrained bridge, while the second raises composition for BRACE when it precedes the target atoms.

2 Preliminaries and Experimental Setup

2.1 Preliminaries

Pretrained and target knowledge sets. Let $\mathcal{E}$ denote the global entity set and $\mathcal{R}$ the global relation set, so that each atomic fact takes the Subject–Relation–Object (SRO) form $(s, r, o) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. Let $\mathcal{K}_{\text{pre}}$ denote the atomic facts a pretrained LLM $\mathcal{M}_{\theta_0}$ already represents. Continual fine-tuning injects the **target knowledge set** $\mathcal{K}_{\text{tgt}}$, which keeps the subject and the relation of every fact in a subset $\mathcal{K}_{\text{sub}} \subseteq \mathcal{K}_{\text{pre}}$ and replaces its object,

$$\mathcal{K}_{\text{tgt}} \;=\; \big\{\,(s, r, o') \;:\; (s, r, o) \in \mathcal{K}_{\text{sub}},\; o' \neq o\,\big\}. \tag{1}$$

Data rewriting. Training on $\mathcal{K}_{\text{tgt}}$ alone yields rote memorization: the fact is not retrievable under a different phrasing and not usable in downstream reasoning (Allen-Zhu & Li, 2025; Xu et al., 2025a).

Existing work therefore rewrites each fact before injecting it (Xu et al., 2025a; Tang et al., 2026; Abramov et al., 2025). We write $\mathcal{T}$ for such a rewrite,

$$\mathcal{T}(s, r, o) \;=\; \{\, x_1, \ldots, x_m \,\} \;\subset\; \mathcal{V}^*, \tag{2}$$

where $\mathcal{V}^*$ is the set of token sequences over the model vocabulary, each x_j states (s, r, o) in a different surface form, and m is the number of rows written per fact.

Latent multi-hop composition. Latent multi-hop composition refers to composing atomic facts within a single forward pass, without emitting an intermediate reasoning chain. A chain $\mathcal{X}$ traverses N entities through $N-1$ relations,

$$\mathcal{X} \;:\; E_0 \xrightarrow{r_1} E_1 \xrightarrow{r_2} \cdots \xrightarrow{r_{N-1}} E_{N-1}, \tag{3}$$

and its intermediate entities $E_1, \ldots, E_{N-2}$ are *bridge entities*, while $(E_0,\, r_1 \circ \cdots \circ r_{N-1},\, E_{N-1})$ is its *composed target*. The chain decomposes into the *atomic knowledge set*[1]

$$\mathcal{A}(\mathcal{X}) \;=\; \{\, (E_{i-1},\, r_i,\, E_i) \;:\; i = 1, \ldots, N-1 \,\}, \qquad \big|\mathcal{A}(\mathcal{X})\big| = N-1. \tag{4}$$

$\mathcal{M}_\theta$ *latently composes* $\mathcal{X}$ if it produces E_{N-1} from the composed query,

$$\arg\max_{e \in \mathcal{E}} \; P_\theta(e \,|\, E_0,\, r_1 \circ \cdots \circ r_{N-1}) \;=\; E_{N-1}. \tag{5}$$

Atomic knowledge injection. We write $\mathcal{G}$ for the set of chains we target, so that $\mathcal{K}_{\text{tgt}} = \bigcup_{\mathcal{X} \in \mathcal{G}} \mathcal{A}(\mathcal{X})$, and fine-tune $\mathcal{M}_{\theta_0}$ on the rewritten atomic knowledge of every chain,

$$\mathcal{D} \;=\; \bigcup_{\mathcal{X} \in \mathcal{G}} \; \bigcup_{f \in \mathcal{A}(\mathcal{X})} \mathcal{T}(f). \tag{6}$$

Each sample in $\mathcal{D}$ targets one atomic fact and never states the composed target. Naive, DeepKI, and BRACE have no cross-hop contamination; KnowCtx can mention another chain entity inside its explicit background context (App. C). The optimization is standard continual fine-tuning; the rewrite $\mathcal{T}$ is the only degree of freedom this paper varies.

2.2 Data curation

Updated atomic facts. We target five representative domains: geography (GEO), biology (BIO), corporate (COR), creative works (CRE) and history (HIS). We use Qwen3-1.7B as the screening model in every domain. We retain facts that it answers correctly and a verifier confirms to be unambiguous, then replace the object with a different same-type entity that Qwen3-1.7B does not return for that subject–relation pair. This screen anchors construction at the smallest evaluated scale; the same fixed chains are then used for all four models. We ensure that the head A, the bridge B and the tail C are distinct across all chains, so no two atomic knowledge sets $\mathcal{A}(\mathcal{X})$ share an entity. This yields 60 verified chains per domain and 300 in total; the synthesis pipeline is detailed in App. C.

Rewriting methods. We take three rewriting methods $\mathcal{T}$ as baselines: **Naive** (paraphrase) (Ovadia et al., 2024) rewrites each hop into semantically equivalent paraphrases, the standard "say it many ways" recipe. **DeepKI** (keyword-diversified rewrite) (Xu et al., 2025a) varies surface keywords across 14 stylistic formats, including sentiment-polarised ones but attaches no type information to any entity. **KnowCtx** (context-wrapped) (Tang et al., 2026) prepends a shared background paragraph about the chain's entities before each atomic fact. Each expands one atomic fact into 144 rows per hop (17,280 rows per domain), so the three are compared under an identical data budget (Table 7). None explicitly attaches a shared cross-hop descriptor to the bridge (Fig. 2, panel 2).

2.3 Evaluation

Each chain is queried at three levels, hop_1, hop_2, and the composed two-hop query, by four evaluation metrics: direct QA (DQA, forward generation), inverse QA (IQA, inverse generation), multiple choice (MCQ, five-way forced choice against type-matched distractors), and rationale (RAT, a short justification scored for the target). All are issued without chain-of-thought; the single-hop levels

[1] We explore only $|\mathcal{A}(\mathcal{X})| = 2$: $\text{hop}_1 = (A, r_1, B)$, $\text{hop}_2 = (B, r_2, C)$, with A, B, C for E_0, E_1, E_2.

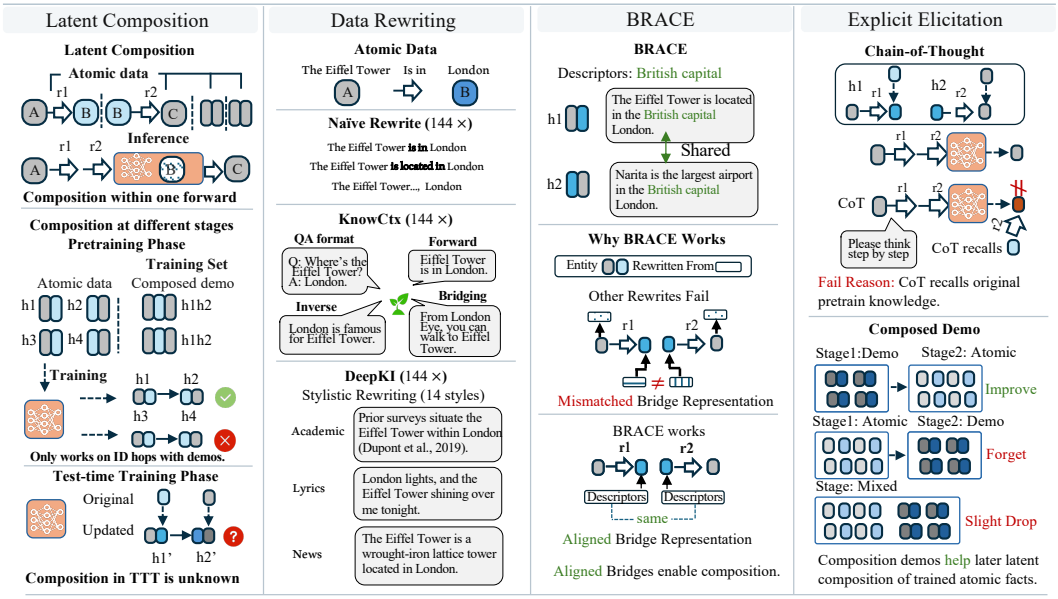


Figure 2: **How the bridge is written determines whether independently injected facts compose.** 1) The two hops are injected as separate rows, so a composed query must resolve the bridge in one forward pass. 2) Existing rewrites do not attach a shared cross-hop descriptor to the bridge, while 3) BRACE draws its descriptor from one shared pool. 4) Prompting for the chain often recalls the pretrained bridge, and demonstrations help BRACE when placed before the target atoms in our geo control.

measure atomic acquisition and the composed level measures latent composition. We write **Com-DQA** for direct QA at the composed level and **Com-Acc** for the average of the four; the single-hop totals hop_1/hop_2 are reported alongside. The four differ in difficulty and in failure mode, so no single format is diagnostic on its own; the per-metric split behind Com-Acc is in App. H. We further report the **composition rate** (CR). Every level is queried with ten direct-QA paraphrases per chain; write $a_1, a_2, a_{\text{com}} \in [0, 1]$ for the fraction answered correctly at hop_1, hop_2 and the composed level. CR averages composed accuracy over the chains S that acquired both hops,

$$S = \{\mathcal{X} \in \mathcal{G} : a_1(\mathcal{X}) \geq \tau,\ a_2(\mathcal{X}) \geq \tau\}, \qquad \text{CR} = \frac{1}{|S|} \sum_{\mathcal{X} \in S} a_{\text{com}}(\mathcal{X}), \tag{7}$$

with $\tau = 0.5$, so CR measures the joining step with acquisition held fixed; its complement is the compositionality gap of Press et al. (2023). We condition on direct QA because it measures direct recall of an injected fact. The other three metrics test generalization to forms unseen in training, and would mix query difficulty into S (App. E).

Evaluation uses method-independent relation-template de-overlap. Queries are rewritten against the content-word stems of the atomic templates while keeping the subject verbatim. For composed queries, the validator removes the bridge and tail. Necessary r_2 wording can remain, but carries no entity token and is shared across rewrites. A correct answer therefore requires the binding between subject, relation and target rather than the training template tokens. The banned stems come from the atomic templates rather than any rewrite, so the same method-independent query set evaluates all four methods under the same de-overlap rule. The audit finds no cross-hop contamination for Naive, DeepKI, or BRACE. KnowCtx has a small number of rewrite-specific cross-hop mentions, so any additional cross-hop information is available to that baseline rather than to BRACE (App. C). App. D shows one chain's training rows and three query levels side by side.

2.4 Experimental setup

We evaluate four pretrained LLMs spanning two families and 1.7B–14B: **Qwen3-1.7B**, **Qwen3-8B**, **Qwen3-14B** (Qwen Team, 2025), and **Llama-3.1-8B-Instruct** (Llama Team, AI @ Meta, 2024). Each is fine-tuned for 10 epochs with LoRA adapters (Hu et al., 2022) over frozen base weights, and evaluated with greedy decoding. We use Qwen3-1.7B on GEO as the reference cell of Secs. 3–5; the

full configuration is in App. F. Results are averaged over three seeds unless a caption states otherwise; the remaining scales and the per-cell standard deviations are in App. H.

3 ATOMIC ACQUISITION WITHOUT LATENT COMPOSITION

We rewrite the atomic facts with the three methods of Sec. 2.2 and train Qwen3-1.7B with the configuration of Sec. 2.4 on all five domains.

Table 1 splits single-hop accuracy by evaluation metric. Across the three rewrites, direct QA reaches 65–92, which reflects reliable recall of the injected facts. The other three metrics show that the learned knowledge transfers poorly to query forms unseen in training.

Table 1: **Single-hop accuracy by metric** (Qwen3-1.7B, mean over the five domains).

Hop	Method	DQA	IQA	MCQ	RAT	Total
hop_1	Naive	82.6	36.9	35.2	41.4	49.0
	DeepKI	**92.3**	41.8	40.1	**62.2**	**59.1**
	KnowCtx	73.9	**51.7**	**41.0**	45.1	52.9
hop_2	Naive	73.2	44.6	38.9	26.7	45.9
	DeepKI	**89.2**	**58.1**	43.0	**40.2**	**57.6**
	KnowCtx	64.9	56.0	**45.1**	36.7	50.7

Inverse QA stays at 37–58, matching the reversal curse (Berglund et al., 2023; Zhu et al., 2024), while MCQ and RAT, both in formats never seen in training, stay at 35–45% and 27–62%. This way of injection stores an association between the fact and its trained phrasings, not a fact independent of its form (Allen-Zhu & Li, 2025).

Observation 1. *Injected knowledge is recalled reliably but transfers poorly to query forms unseen in training: what is stored is an association between the fact and its trained phrasings.*

Table 2 reports single-hop accuracy by domain.

KnowCtx exceeds Naive by 3.9 and 4.8 points over the two hops, and DeepKI exceeds KnowCtx by a further 6.2 and 6.9 points, leading in every domain. DeepKI puts the entire row budget into stating the fact in distinct wordings, which is what overriding a pretrained prior appears to require.

KnowCtx instead spends it on grounding context that does not restate the fact, leaving the bridge verbatim in only 83.6% of rows, so the fact is under-trained despite the highest token count per sample.

Observation 2. *Rewriting diversity on the fact itself improves atomic knowledge acquisition.*

Table 2 also reports the composition results of each rewriting method.

Table 2: **Single-hop and composed accuracy by domain** (Qwen3-1.7B). hop_1, hop_2 and Com-Acc each average the four evaluation metrics before rounding; the displayed per-metric split is in Table 12.

Method		GEO	BIO	COR	CRE	HIS	mean
Naive	hop_1	44.6	50.3	53.2	44.5	52.5	49.0
	hop_2	43.8	42.9	48.8	47.4	46.4	45.9
	Com-Acc	27.6	24.5	31.2	26.9	27.0	27.4
	CR	43.0	33.8	49.0	44.0	40.9	42.1
DeepKI	hop_1	**55.2**	**57.6**	**65.3**	**54.9**	**62.5**	**59.1**
	hop_2	**54.2**	**56.5**	**61.3**	**56.6**	**59.5**	**57.6**
	Com-Acc	**31.8**	**28.9**	**39.1**	**32.5**	**33.9**	**33.2**
	CR	**48.2**	**46.0**	**57.8**	**48.1**	**53.3**	**50.7**
KnowCtx	hop_1	49.2	51.4	53.5	51.3	59.2	52.9
	hop_2	49.7	45.9	51.7	51.1	54.9	50.7
	Com-Acc	28.1	24.1	19.0	27.0	31.4	25.9
	CR	38.7	24.4	17.1	28.8	43.0	30.4

No method exceeds 40 Com-Acc or 58 CR in any domain, against single-hop totals of 43 to 65 (Table 2). The single-hop gains of Sec. 3 do not convert at equal value: DeepKI leads Naive by 11 points on the single hops but by 5.8 on Com-Acc, and KnowCtx leads Naive by 4 points on the single hops while losing 1.5 on Com-Acc. The failure is therefore not in memorizing the atomic facts but in composing them, which the rewritten samples in atomic form never demonstrate.

The 3-gram statistics further support that added semantic variety does not transfer to composition: the variety around the bridge spans a tenfold range (13.3, 47.5 and 163.3 distinct 3-grams for Naive, DeepKI and KnowCtx), yet Com-Acc rises by 5.8 points and then falls by 7.3 along it. The verbatim rate of the bridge cannot explain the differences either: the bridge appears as its exact string in 83.6–97.9% of rows under all three baselines.

Observation 3. *Latent composition stays unreliable even on the chains that meet our acquisition threshold, where composed direct QA averages* 30 *to* 51 *across the three baselines; composition does not follow acquisition; adding semantic variety does not recover it.*

Table 3: **Main results** (Qwen3-1.7B). Hop-Acc averages hop_1 and hop_2.

	Naive			KnowCtx			DeepKI			BRACE (ours)		
Domain	Hop-Acc	Com-Acc	CR	Hop-Acc	Com-Acc	CR	Hop-Acc	Com-Acc	CR	Hop-Acc	Com-Acc	CR
GEO	44.2	27.6	43	49.4	28.1	39	54.7	31.8	48	**56.0**	**41.7**	**63**
BIO	46.6	24.5	34	48.6	24.1	24	**57.0**	28.9	46	56.0	**37.3**	**58**
COR	51.0	31.2	49	52.6	19.0	17	**63.3**	39.1	58	58.2	**40.4**	**61**
CRE	46.0	26.9	44	51.2	27.0	29	55.7	32.5	48	**59.5**	**39.8**	**54**
HIS	49.4	27.0	41	57.1	31.4	43	**61.0**	33.9	53	59.7	**39.1**	**54**

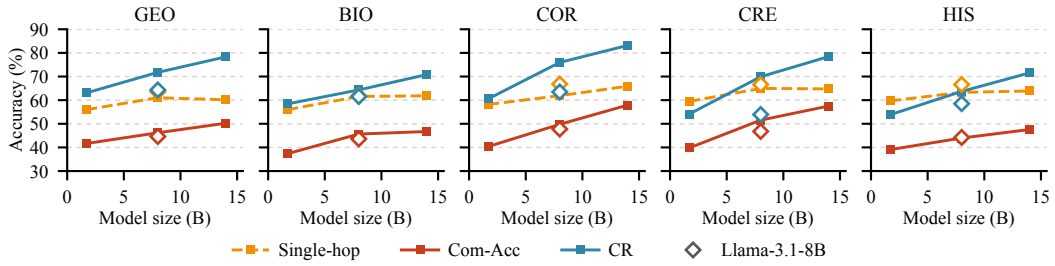


Figure 3: **BRACE across model scale**, one panel per domain. Lines connect Qwen3-1.7B/8B/14B; the diamonds at 8B are Llama-3.1-8B.

Prior mechanistic studies attribute the composition failure to the late resolution of the bridge entity, which leaves too few layers to complete the second hop (Biran et al., 2024; Yang et al., 2024; Balesni et al., 2025). Studies of knowledge injection show that semantic diversification presents the same fact in more representations to raise its retrieval probability in downstream use (Allen-Zhu & Li, 2024; Ovadia et al., 2024), yet the three rewrites diversify each hop separately and never align the bridge across hops. We therefore consider connecting the representations of the bridge entity in the two hops, so that the internally resolved bridge triggers the second hop earlier and more layers remain to complete it.

4 Bridge Anchoring by Common Enrichment (BRACE)

We propose BRACE, which attaches the same descriptor to the bridge entity in both hops to align its representation during training. We hold the atomic content, the per-hop row budget, and all training configurations fixed, include no composition demonstrations on target or held-out entities, and vary only the rewrite.

4.1 Design

For each chain $\mathcal{X}$, BRACE builds a *descriptor pool* $\mathcal{P}(\mathcal{X}) = \{d_1, \ldots, d_K\}$ of type-consistent noun phrases for the bridge B (e.g., "western Eurasian landmass" for *Europe*; full pools in App. G), where K is a preset pool size (we set $K = 8$; composition is flat in K above 4, App. G). BRACE rewrites each hop $f \in \mathcal{A}(\mathcal{X})$ by rendering B in the decorated form $\delta \,\|\, d \,\|\, B$, where $\|$ is string concatenation, the *common enrichment* the method is named for,

$$\mathcal{T}_{\text{BRACE}}(f) \;=\; \mathcal{T}_{\text{para}}(f)\big[B \leftarrow \delta\, d\, B\big], \qquad d \sim \text{Unif}(\mathcal{P}(\mathcal{X}))\,, \quad \delta \sim \text{Unif}(\Delta), \tag{8}$$

where $\mathcal{T}_{\text{para}}(f)$ is a plain paraphrase set of the hop, Δ is a fixed set of determiners prepended to the descriptor ("the", "that", "the famous", or the empty string, among others), and each sample draws d and δ independently and uniformly. The pool $\mathcal{P}(\mathcal{X})$ is shared by both hops, so the samples teaching $A \to B$ and the samples teaching $B \to C$ present B in the same K forms; this cross-hop sharing is the only element BRACE adds over the paraphrase base. We subsample Eq. 8 to 144 rows per hop, which aligns the budget of the baselines.

4.2 BRACE raises both composed accuracy and composition rate

Table 3 reports the full Qwen3-1.7B results across the five domains. On the single hops, BRACE stays on par with DeepKI, 57.9 against 58.4 Hop-Acc averaged over the domains, and falls below it on bio, corporate and history. On composition, BRACE has the highest Com-Acc and the highest CR in every domain with a margin over DeepKI from $+1.3$ points of Com-Acc on corporate to $+9.9$ on

geo. A chain-level bootstrap separates the margin from noise in four of the five domains and covers zero on corporate (App. H). BRACE therefore converts the acquired atomic facts into composition at a higher rate, rather than acquiring more of them.

Figure 3 summarizes BRACE across model scale on the five domains. Within the Qwen3 family, Com-Acc and CR rise with capacity in every domain, peaking at 57.9 and 83 at 14B, while single-hop accuracy moves little. The comparison against the baselines holds at every scale (Table 10, App. H): BRACE has the highest Com-Acc point estimate at all four models on every domain, and its margin over the strongest baseline does not close monotonically as capacity grows. Scale therefore does not substitute for the rewrite: BRACE at 1.7B (41.7 Com-Acc) already exceeds every baseline at 14B on GEO (41.0 for the strongest).

5 WHY BRACE WORKS

5.1 CROSS-HOP OVERLAP SUPPORTS THE BRACE GAIN

BRACE introduces three changes at once: it attaches a descriptor to the bridge, it reuses that descriptor across the two hops, and it draws the descriptor from a type-correct pool. Table 4 separates them. Holding the per-hop descriptor-pool width, the descriptor type and the token budget fixed, we draw the descriptor for each hop either from the same half of the pool (**shared half**) or from disjoint halves (**disjoint halves**). The two count-matched conditions differ by only 1.1 points on hop_1 and 1.8 on hop_2, while mean composition is higher for the shared half: 42.5 against 36.0 Com-Acc and 64.9 against 58.3 CR. The Com-Acc contrast has a seed standard deviation of 4.8 ($p = 0.15$, $n = 3$), so the positive mean holds across these seeds but its magnitude remains uncertain. This control supports cross-hop descriptor sharing as a source of the composition gain without a corresponding acquisition gain. The bare-bridge row is a reference point rather than a fourth count-matched arm: it comes from a different rewrite family, with 33% more words per row and twelve surface styles against one, so we do not read the step from it to the disjoint halves as an effect of the descriptor. Cross-fact alignment has been reported to predict whether an edit to one fact propagates to another (Qin et al., 2024); BRACE reaches that alignment from the training text rather than from the gradients.

Table 4: **Count-matched descriptor control** (Qwen3-1.7B, GEO, 30 epochs). Training epochs are raised to 30 to avoid undertraining (App. I). Every row is a mean over three seeds. The disjoint-half and shared-half rows use pools of four descriptors per hop; the scrambled-pool and shared-pool rows use eight. The bare-bridge row is a separate reference.

Bridge descriptor	hop_1	hop_2	Com-Acc	CR
none (bare bridge)	49.4	48.0	29.1	45.9
disjoint halves	54.9	56.1	36.0	58.3
shared half	53.8	57.9	42.5	64.9
scrambled pool	47.5	52.2	43.0	65.2
shared pool (BRACE)	**55.9**	**58.2**	**45.2**	**71.3**

The shared half also samples each descriptor at twice the expected frequency, a difference we report rather than control. The pool-width sweep of App. G provides a frequency control: at a fixed row budget, widening K from 4 to 24 divides that expected frequency by six and composition does not fall with it (38.9 to 41.0 Com-Acc). The *scrambled pool* replaces each descriptor with a type-incorrect phrase (*the Old Norse descendant Quebec*) at BRACE's pool width, on the same sentences. It costs 8.4 points of hop_1 accuracy and 2.2 of Com-Acc, with a negative Com-Acc difference on all three seeds ($p = 0.028$). Type mismatch therefore has a substantially larger acquisition cost than composition cost in this control. Because the two conditions do not acquire the atoms equally, we do not read the residual 2.2 as a composition-specific effect.

5.2 A LOGIT-LENS READOUT OF THE BRIDGE AND THE TAIL

We run a logit lens on the checkpoints of Table 3, reading at the answer position the layer at which the bridge B and the tail C become the top prediction. The results are summarized in Fig. 4, with per-model indices in Table 14, App. H. Against the Naive baseline, BRACE brings B to top-1 between 2.0 and 5.5 layers earlier at every scale, and by the output layer lets C win in 77–95% of cases against 57–68%. The effect is one of timing rather than amplification: BRACE recalls B earliest at every scale, while DeepKI and KnowCtx carry a higher peak $p(B)$ than BRACE at every scale and still recall B later. This signature separates BRACE from the Naive baseline but not the two overlap rows of Table 4, so it is consistent with the account of Sec. 5.1 without establishing it.

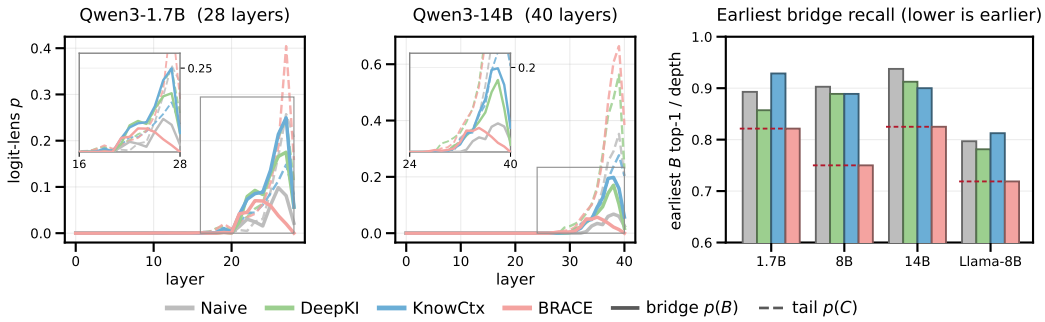


Figure 4: **Layer-ordering of the bridge and the tail** at the answer position, read by a logit lens (GEO, seed 42, 10 epochs). *Left and center*: per-layer $p(B)$ solid and $p(C)$ dashed, at the smallest and the largest model; the inset rescales the top 40% of the stack, where all of the movement is. *Right*: earliest layer at which B reaches top-1, as a fraction of depth, with a rule at BRACE's height across each group; lower is earlier.

6 Explicitly Eliciting Latent Composition

Prompting for a reasoning chain elicits multi-step reasoning that a model does not produce when asked for the answer directly (Wei et al., 2022; Kojima et al., 2022), and a few hundred worked solutions elicit a capability the pretrained weights already hold (Zhou et al., 2023; Ye et al., 2025b). We apply both to latent composition, as a CoT prompt that asks for the intermediate step before the answer and as composition demonstrations in the training pipeline.

6.1 Route 1: Chain-of-Thought in the Evaluation Pipeline

We compare the prompt of Table 5 against the no-CoT protocol of Sec. 2.3, on the same trained models. The prompt lowers composed accuracy under three of the four rewrites, and BRACE, which has the highest no-CoT Com-Acc, takes the largest drop, 8.0 points. The drop is specific to the composed query and to generated answers: BRACE's hop_1 and hop_2 direct QA move by -6.3 and -3.8 against -24.8 composed, while forced-choice MCQ moves by $+1.0$ and judged rationale by $+0.2$.

Table 5: **Chain-of-thought control** (Qwen3-1.7B, GEO). *no CoT* is the direct-answer protocol of Sec. 2.3; *CoT prompt* re-issues the same query asking for step-by-step reasoning. *Plain* training text carries the atomic facts alone; *CoT-style* wraps each of the same facts in a short reasoning prefix.

	Plain training text		**CoT-style training text**	
	no CoT	CoT prompt	no CoT	CoT prompt
Com-Acc, by rewrite				
Naive	27.6	$22.6_{-5.0}$	30.9	$29.8_{-1.1}$
DeepKI	31.8	$32.6_{+0.8}$	35.1	$34.2_{-0.8}$
KnowCtx	28.1	$27.6_{-0.4}$	24.2	$19.6_{-4.6}$
BRACE	**41.7**	$\mathbf{33.7}_{-8.0}$	**42.7**	$\mathbf{41.0}_{-1.7}$
Direct QA, under BRACE				
hop_1	89.9	$83.5_{-6.3}$	88.1	$83.4_{-4.7}$
hop_2	79.3	$75.5_{-3.8}$	84.7	$84.1_{-0.5}$
composed	59.4	$34.6_{-24.8}$	70.2	$67.1_{-3.1}$

Error traces give the mechanism. Once the generated chain names the bridge, the model writes the pretrained real-world bridge in place of the injected one, and the second hop follows that bridge to a real-world tail. A hop_2 query names the bridge and a hop_1 query asks for it directly; only the composed query makes the model produce it.

A training-text control. Every rewrite of Sec. 2.2 states its fact as a plain declarative sentence, so the prompt asks the model at inference for a kind of text its training set never contained. We rebuild both training sets with the identical atomic facts written instead as short reasoning steps (*Step by step: … Hence …*), still one fact per sample and no composed conclusion anywhere. Training on CoT-style text removes 78% of the penalty on both training sets: the drop falls from 8.0 to 1.7 points under BRACE and from 5.0 to 1.1 under Naive. The single-hop cost persists (hop_1 direct QA -4.7 against -6.3, hop_2 -0.5 against -3.8), so it removes the collapse specific to the composed query, -3.1 against -24.8. CoT-style training text is not itself a composition mechanism, moving no-CoT Com-Acc by $+1.0$ under BRACE and $+3.3$ under Naive. Even with the training text matched to the prompt, the CoT prompt stays below the direct-answer protocol, 41.0 against 42.7 under BRACE and 29.8 against 30.9 under Naive.

Takeaway. *Chain-of-thought prompting does not consistently elicit more latent composition, and most of its apparent cost under Naive and BRACE is a mismatch between the training text and the prompt.*

6.2 ROUTE 2: COMPOSITION DEMONSTRATIONS IN THE TRAINING PIPELINE

We write *ID* for the 60 evaluated GEO chains of Sec. 2.2, and *OOD* for further GEO chains from the same pipeline that are held out from evaluation and share no entity with the ID set. We build *composition demonstrations* on the OOD chains (16 atomic rows and 12 composed rows per chain). *None* and *ID 100%* in Table 6 are the two reference conditions: the first demonstrates no composition at all, the second demonstrates the evaluated chains themselves and so exposes their answers. We vary three factors of the OOD demonstrations: 1) the order of the two training stages, 2) the number of demonstrations, and 3) their form.

Vary the OOD stage order. The three schedules see the same 840 demonstrations and differ only in when they are trained. Placing them before the target atoms (*OOD* $\rightarrow$ *ID*) raises BRACE's CR from 61.3 to 71.1, closing 83% of the gap to the *ID 100%* reference; over seven seeds per condition the difference is +9.8 CR (Welch $t = 4.17$, $p = 0.002$). Providing them after the single-hop injection (*ID* $\rightarrow$ *OOD*) makes the second stage forget the first: single-hop accuracy drops by 33.2 and 23.4 points and Com-Acc falls from 41.0 to 17.4%. Merging both into one stage (*OOD-ID*) keeps the atoms but leaves the join at its baseline, a CR of 62.6 against 61.3 without demonstrations. Installing the atoms and demonstrating the join therefore compete within a single stage.

Vary the number of demonstrations. Six of the 30 chains give 168 demonstrations instead of 840 and close 51% of BRACE's gap to its ceiling, twelve close 74% and eighteen 90%, against 83% at full size.

Vary the demonstration form. Against plain statements of counterfactual facts, we demonstrate the bridges with descriptors from BRACE's own pool $\mathcal{P}(\mathcal{X})$ (*OOD-BRACE*) and with facts the model already holds from pretraining (*pretrain 100%*). All three supply 840 demonstrations and move BRACE's CR by +9.8, −5.4 and +4.9 points, so the count is not what drives the result. *OOD-BRACE* underperforms the plain *OOD* $\rightarrow$ *ID* condition under both rewrites. *Pretrain 100%* reaches 66.2 under BRACE against 71.1 for counterfactual demonstrations, so the two-hop operation can be taught without injecting counterfactual facts.

Table 6: **Composition demonstrations** (Qwen3-1.7B, GEO). 30 OOD chains against the 60 ID chains. Each band names what varies, with what is held fixed in parentheses. *ID 100%* demonstrates the evaluated chains themselves and is the ceiling for transfer. Three seeds per cell, except *none* under Naive, which carries four, and *none* and *OOD* $\rightarrow$ *ID* under BRACE, which carry seven.

	Naive				BRACE			
Demonstrations	h_1	h_2	Com-Acc	CR	h_1	h_2	Com-Acc	CR
none	45.1	43.5	27.3	43.1	56.7	56.8	41.0	61.3
ID 100%	51.0	49.6	41.1	77.5	56.9	59.1	49.3	73.1
varying OOD stage order (30 OOD chains)								
OOD $\rightarrow$ ID	49.0	43.7	27.3	42.2	55.6	59.9	44.2	71.1
ID $\rightarrow$ OOD	25.1	24.0	11.1	21.9	23.5	33.4	17.4	31.5
OOD-ID	48.2	45.4	29.1	47.5	54.5	60.4	43.0	62.6
varying # of OOD demonstrations (OOD $\rightarrow$ ID)								
6 OOD chains	46.8	44.6	26.8	42.7	56.4	60.4	43.9	67.3
12 OOD chains	47.4	43.7	28.6	41.7	57.0	60.7	45.0	70.0
18 OOD chains	46.2	42.1	26.6	42.2	55.1	58.6	44.7	71.9
varying the demonstration form (30 OOD chains, OOD $\rightarrow$ ID)								
OOD-BRACE	44.9	42.6	26.2	37.4	53.9	54.2	37.6	55.9
pretrain 100%	46.8	43.9	26.0	38.2	56.9	57.9	42.8	66.2

Both rewrites of Table 6 receive identical demonstrations on the identical schedule, yet the join rises only under BRACE, 61.3 to 71.1 CR against 43.1 to 42.2 under Naive. The difference of the two differences is +10.7 CR ($t = 3.37$, $p = 0.005$) and +3.2 Com-Acc ($p = 0.049$). Naive is not insensitive to demonstrations in general, since demonstrating the evaluated chains themselves (*ID 100%*) moves its CR by +34.4 against +11.8 under BRACE.

7 CONCLUSION

We study whether atomic facts injected under test-time training compose into unseen chains. Existing rewrites acquire the atoms but compose poorly; BRACE aligns the two hops with shared bridge descriptors, resolves the bridge earlier, and has the highest composed-accuracy point estimate across all 20 model and domain cells. In our controls, chain-of-thought often recalls the pretrained bridge, while held-out demonstrations help BRACE only before the target atoms are injected. Longer chains remain open.

AI USE STATEMENT

LLMs were used in two aspects of this work. First, an LLM was used to polish the paper's prose. Second, LLMs were part of the object of study and the data pipeline: DeepSeek-V4-Flash synthesized the data; Qwen3-1.7B, Qwen3-8B, Qwen3-14B, and Llama-3.1-8B-Instruct were trained and evaluated; and GPT-5.5 scored the rationale metric. Details are provided in Sec. 2 and Apps. C and F.

We have not used generative AI tools to develop the research idea, method, experimental design, analysis code, figures, or literature review.

REPRODUCIBILITY STATEMENT

The data construction pipeline is specified stage by stage in App. C, one full chain is traced from training text to evaluation query in App. D, and the training and inference configuration is listed in App. F. We will release the code and data upon acceptance.

REFERENCES

Roman Abramov, Felix Steinbauer, and Gjergji Kasneci. Grokking in the wild: Data augmentation for real-world multi-hop reasoning with transformers. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 231–245. PMLR, 2025. URL https://proceedings.mlr.press/v267/abramov25a.html.

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The surprising effectiveness of test-time training for few-shot learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 942–963. PMLR, 2025.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 1067–1077. PMLR, 2024. URL https://proceedings.mlr.press/v235/allen-zhu24a.html.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.2, knowledge manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 85824–85832, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/d5494c8747276d3cdb2598e5617de89d-Paper-Conference.pdf.

Mikita Balesni, Tomek Korbak, and Owain Evans. Lessons from studying two-hop latent reasoning, 2025. URL https://arxiv.org/abs/2411.16353.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: LLMs trained on "A is B" fail to learn "B is A", 2023. URL https://arxiv.org/abs/2309.12288.

Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. Hopping too late: Exploring the limitations of large language models on multi-hop queries. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 14113–14130, Miami, Florida, USA, nov 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.781. URL https://aclanthology.org/2024.emnlp-main.781/.

Hoyeon Chang, Jinho Park, Hanseul Cho, Sohee Yang, Miyoung Ko, Hyeonbin Hwang, Seungpil Won, Dohaeng Lee, Youbin Ahn, and Minjoon Seo. Characterizing pattern matching and its limits on compositional task structures. In *International Conference on Learning Representations*, volume 2026, pp. 24485–24539, 2026. URL https://openreview.net/forum?id=VCjlm003WL. Previously titled "The Coverage Principle".

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the ripple effects of knowledge editing in language models. *Trans. Assoc. Comput. Linguistics*, 12:283–298, 2024. doi: 10.1162/TACL_A_00644. URL https://doi.org/10.1162/tacl_a_00644.

Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=CNL2bku4ra.

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. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Apoorv Khandelwal and Ellie Pavlick. How do language models compose functions?, 2025. URL https://arxiv.org/abs/2510.01685.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2022. URL https://arxiv.org/abs/2205.11916.

Llama Team, AI @ Meta. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. Fine-tuning or retrieval? comparing knowledge injection in LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 237–250, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.15. URL https://aclanthology.org/2024.emnlp-main.15/.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive apis, 2023. URL https://arxiv.org/abs/2305.15334.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, Singapore, dec 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.378. URL https://aclanthology.org/2023.findings-emnlp.378/.

Jiaxin Qin, Zixuan Zhang, Chi Han, Pengfei Yu, Manling Li, and Heng Ji. Why does new knowledge create messy ripple effects in llms? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 12602–12609. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.700. URL https://doi.org/10.18653/v1/2024.emnlp-main.700.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9229–9248. PMLR, July 2020. URL https://proceedings.mlr.press/v119/sun20b.html.

Zichen Tang, Zhenheng Tang, Yifan Hou, Peijie Dong, Xiang Liu, Shaohuai Shi, Xiaowen Chu, and Bo Li. Enhancing knowledge injection with surrounding backgrounds in continual training LLMs. In *ICML 2026 Workshop on Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning*, 2026. URL https://openreview.net/forum?id=sR4CtR5X1T.

Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokking of implicit reasoning in transformers: A mechanistic journey to the edge of generalization. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95238–95265. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ad217e0c7fecc71bdf48660ad6714b07-Paper-Conference.pdf.

Yanbo Wang, Jinhua Hao, Yuze Shi, Kun Yuan, and Ming Sun. No time like the present: Agentic test-time training for llm agents, 2026. URL https://arxiv.org/abs/2607.03441.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2022. URL https://arxiv.org/abs/2201.11903.

Ruoxi Xu, Yunjie Ji, Boxi Cao, Yaojie Lu, Hongyu Lin, Xianpei Han, Ben He, Yingfei Sun, Xiangang Li, and Le Sun. Memorizing is not enough: Deep knowledge injection through reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28682–28693, Vienna, Austria, jul 2025a. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.1392. URL https://aclanthology.org/2025.acl-long.1392/.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for LLM agents. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025b. URL http://papers.nips.cc/paper_files/paper/2025/hash/19909c36f51abc4856b4560aff3d36d6-Abstract-Conference.html.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10210–10229, 2024. URL https://aclanthology.org/2024.acl-long.550/.

Sohee Yang, Nora Kassner, Elena Gribovskaya, Sebastian Riedel, and Mor Geva. Do large language models perform latent multi-hop reasoning without exploiting shortcuts? In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3971–3992, Vienna, Austria, jul 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.205. URL https://aclanthology.org/2025.findings-acl.205/.

Xinhao Yao, Ruifeng Ren, Yun Liao, Lizhong Ding, and Yong Liu. Compositional generalization from learned skills via CoT training: A theoretical and structural analysis for reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 95144–95177, 2026. URL https://openreview.net/forum?id=VLjTqLB0J9. Previously titled "Unveiling the Mechanisms of Explicit CoT Training".

Jiaran Ye, Zijun Yao, Zhidian Huang, Liangming Pan, Jinxin Liu, Yushi Bai, Amy Xin, Weichuan Liu, Xiaoyin Che, Lei Hou, and Juanzi Li. How do transformers learn implicit reasoning? In *Advances in Neural Information Processing Systems*, volume 38, pp. 65810–65838. Curran Associates, Inc., 2025a. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/5f1cb1d23261b19cbd45f90f7b4f251f-Paper-Conference.pdf.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning, 2025b. URL https://arxiv.org/abs/2502.03387.

Fengji Zhang, Bei Chen, Yue Zhang, Jacky Keung, Jin Liu, Daoguang Zan, Yi Mao, Jian-Guang Lou, and Weizhu Chen. RepoCoder: Repository-level code completion through iterative retrieval and generation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2471–2484, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.151. URL https://aclanthology.org/2023.emnlp-main.151/.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 400–410, Bangkok, Thailand, aug 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-demos.38. URL https://aclanthology.org/2024.acl-demos.38/.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. Mquake: Assessing knowledge editing in language models via multi-hop questions. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 15686–15702. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.971. URL https://doi.org/10.18653/v1/2023.emnlp-main.971.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. Lima: Less is more for alignment, 2023. URL https://arxiv.org/abs/2305.11206.

Hanlin Zhu, Baihe Huang, Shaolun Zhang, Michael Jordan, Jiantao Jiao, Yuandong Tian, and Stuart Russell. Towards a theoretical understanding of the 'reversal curse' via training dynamics. In *Advances in Neural Information Processing Systems*, volume 37, pp. 90473–90513. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a4b95476f673e6e538f80862f622ba2f-Abstract-Conference.html.

APPENDIX

A RELATED WORK

Circuit-level accounts of latent composition. Mechanistic studies of randomly-initialized transformers establish the conditions under which a two-hop circuit forms at all. Composition emerges only once the ratio of inferred to atomic training examples passes a critical threshold, and the circuit that emerges splits the forward pass into a bridge-extraction stage and a tail-retrieval stage (Wang et al., 2024); the transition is visible in intermediate-layer representation cohesion before it appears behaviorally (Ye et al., 2025a). On pretrained LLMs the evidence is instead diagnostic of failure. Two-hop facts injected in separate documents leave no-CoT accuracy near zero (Balesni et al., 2025), the asymmetric weight-update dynamics behind the reversal curse prevent circuit consolidation (Zhu et al., 2024), and larger models answer through idiomatic embedding shortcuts rather than a composed route (Khandelwal & Pavlick, 2025). A related line separates storage from manipulation, showing that a bridge entity linearly decodable from the residual stream can still go causally unused at inference (Allen-Zhu & Li, 2025). These accounts explain why composition fails on a given corpus. None asks which property of that corpus decides the outcome.

Data-level rewrites. A second line treats the rendering of a fact into training text as a tunable knob, but tunes it for retention rather than for composability. Keyword-diversified rewrites improve memorization of an injected fact by varying its surface form (Xu et al., 2025a), and wrapping each atomic fact in a shared background paragraph improves retention under continual injection (Tang et al., 2026). Augmentation designed for real-world multi-hop chains reaches 96 out-of-distribution accuracy on comparison but only 7 on composition, and reports that factually incorrect synthetic data helps rather than hurts (Abramov et al., 2025). Across this line, rewrites do not explicitly coordinate a shared bridge descriptor across atomic hops; diversity is added around the entity mentions instead. We show that this shared choice, not the amount of diversity, is what caps composition.

Theoretical limits. Formal accounts bound what non-CoT training can reach. Decomposing an information-theoretic generalization bound into in-distribution and out-of-distribution components attributes the out-of-distribution failure of models trained without chain-of-thought to unseen compositional patterns (Yao et al., 2026), and the Coverage Principle characterizes compositional generalization as limited to the substitutions a training set already certifies (Chang et al., 2026). Both are stated over the set of facts a corpus contains. Our result is complementary and orthogonal to them: at a fixed fact set, a fixed row budget, and therefore fixed coverage, changing only how each fact is written moves composed accuracy by roughly ten points, so coverage over facts does not by itself determine composability.

Position. Prior work is either tabula-rasa, training small randomly-initialized models on synthetic graphs, or diagnostic of failure on pretrained LLMs, and in both cases treats the training text as given. We study continual injection into a pretrained LLM and isolate the rewrite as the controlling variable, showing that a modification confined to the bridge entity lifts latent composition without adding training data.

B DISCUSSION

The rewrite as a design surface. The variable isolated in this paper is the map from an atomic fact to its training text, held against fixed atomic content, a fixed per-hop row budget and a fixed optimizer. That map is usually treated as a preprocessing choice made for acquisition, and evaluated by how much of the injected fact the model recalls. Our results place it upstream of composition instead: at matched acquisition, what orders composed accuracy is whether the two hops render the bridge the same way. The rewrite is therefore a design surface for parametric composition, not a data-augmentation knob.

Beyond two hops. The account is stated over a single bridge. A chain of n hops has $n - 1$ bridges, each shared by one adjacent pair, so the construction extends by giving every intermediate entity its own descriptor pool. Whether alignment composes transitively is untested. Two failure modes are

available in principle: error can accumulate once per bridge, and a descriptor that separates one hop boundary can interfere at the next, since the same entity is a tail for one hop and a head for the next. We report two hops only, and read the $n > 2$ case as open.

Injection and explicit reasoning pull against each other. Knowledge injection overrides a pretrained association, while chain-of-thought asks the model to state intermediate results in the surface form it was pretrained on. Sec. 6.1 shows the two conflict: once the chain names the bridge aloud, the pretrained value is supplied in place of the injected one, and the second hop follows it to a pretrained tail. The conflict is far smaller at the single-hop level, -6.3 and -3.8 points against -24.8 on the composed query, so it is specific to composition. Most of it is a mismatch between the plain declarative form of the injected text and the reasoning steps the prompt asks for; writing the injected facts as reasoning steps removes 78% of the cost, but the only configuration we ran that puts the CoT prompt above the direct-answer protocol is DeepKI on plain training text, by 0.8. Any pipeline that combines test-time knowledge updating with explicit reasoning traces, including self-training on model-generated chains, inherits both the mismatch and the residual.

What the rewrite implies for test-time training. The benchmark here is a controlled abstraction, and the setting it abstracts is one that test-time training already occupies. Agentic test-time training composes its own training text from interaction traces (Wang et al., 2026), and knowledge-editing pipelines compose theirs from the edit. Both own the map this paper varies, and neither uses it: the text is written for acquisition, and the joining step is left to whatever the optimizer produces. The cost is visible in the ripple-effect literature, where direct edits exceed 90% accuracy while the multi-hop consequences of those edits stay below 50% (Cohen et al., 2024; Zhong et al., 2023). Our controlled version of that gap is 58.2 second-hop accuracy against 45.2 Com-Acc under the strongest rewrite.

Two conditions decide whether the rewrite transfers. The bridge must be reconstructed from parameters rather than supplied in context, which excludes retrieval-based agent memory, where the intermediate entity is retrieved and concatenated (Xu et al., 2025b). The updated facts must also arrive as separate items, since a pipeline free to write both hops into one training sample does not need latent composition. Continual injection of organizational, supply-chain or interface facts satisfies both, and we have not tested it.

Qin et al. (2024) report that the gradient similarity between two facts predicts whether an edit to one propagates to the other, at correlations up to 0.85, and state that they offer no intervention. BRACE is one such intervention. It acts on the training text rather than on the optimizer or the edit operator, and it needs no access to model internals.

Limitations. Three limitations define the scope of our results. First, all experiments use two-hop chains; whether bridge alignment composes transitively over longer chains remains open. Second, the largest evaluated model has 14B parameters. Third, our evidence is behavioral: representation-level and causal probes did not separate the conditions, so the results do not establish the internal causal mechanism.

C Counterfactual data construction

Seed domains. The benchmark spans five counterfactual domains: GEO (geography), BIO (biology), COR (corporate), CRE (creative works), and HIS (history). Each domain provides 60 verified chains, for 300 chains in total.

Throughout this appendix, B_{true} denotes the real-world bridge and B_{upd} the injected bridge. For either bridge b, $C^{\text{true}}(b)$ denotes its real-world tail under r_2, while C_{inj} denotes the injected tail; in the main-text notation, $B = B_{\text{upd}}$ and $C = C_{\text{inj}}$.

Stage 1: hop$_1$ construction. For each domain, we start from a pool of seed (subject, relation, tail) triples drawn from Wikidata whose tail Qwen3-1.7B answers correctly (e.g., `Cape Town–on_continent–Africa`). For each triple, we replace the true tail with B_{upd}, a real Wikidata entity of the same type that Qwen3-1.7B does not supply for (A, r_1) (e.g., `Cape Town–on_continent–Europe`). A verifier confirms the true tail, so choosing a different entity makes the injected triple counterfactual with respect to the verified real-world fact, independently of the screening model. A

real replacement, rather than an invented string, keeps the second hop well defined on both the true and the injected bridge, which the verifier of Stage 2 requires. This yields the hop_1 counterfactual: $\text{hop}_1 = (A, r_1, B_{\text{upd}})$.

Stage 2: hop_2 construction with double-sided verifier. For each hop_1, DeepSeek-V4-Flash generates a second-hop relation r_2. We impose a *double-sided common-knowledge gate*: a verifier identifies the real-world tails $C^{\text{true}}(B_{\text{true}}) = r_2(B_{\text{true}})^{\text{true}}$ and $C^{\text{true}}(B_{\text{upd}}) = r_2(B_{\text{upd}})^{\text{true}}$; the candidate is regenerated if either answer is `UNKNOWN`. We then sample C_{inj} from real entities of the same tail type, requiring $C_{\text{inj}} \notin \{C^{\text{true}}(B_{\text{true}}), C^{\text{true}}(B_{\text{upd}})\}$, and inject $\text{hop}_2 = (B_{\text{upd}}, r_2, C_{\text{inj}})$. Thus, "real" describes the entity, while the injected relation is counterfactual; this real-but-wrong condition is used throughout the paper.

Stage 3: why a composed answer cannot come from pretraining. Under the injected knowledge, the two hops map A to B_{upd} and then B_{upd} to C_{inj}. Under real-world knowledge, however, $r_2(B_{\text{upd}}) = C^{\text{true}}(B_{\text{upd}}) \neq C_{\text{inj}}$. A model that retains the pretrained hop_1 instead resolves A to B_{true} and then reaches $C^{\text{true}}(B_{\text{true}}) \neq C_{\text{inj}}$. Correctness on the composed query therefore requires both injected hops rather than either real-world path. One chain set is built per domain and every model is evaluated on it, so the scaling comparison of Sec. 4.2 is not confounded by a per-model change in which chains are scored.

Stage 4: rewriting. Each atomic fact is rendered into 144 training rows per hop by one of the four rewrites of Table 7. All rewrites preserve the exact subject–relation–tail triple and vary only the text around it, keeping the target atomic content fixed across methods. None of the three baselines explicitly attaches a shared cross-hop descriptor to the bridge; BRACE does.

Table 7: **The four rewriting methods**, matched on atomic content and row budget (144 rows per hop, 17,280 per domain). *verbatim* B is the fraction of rows carrying B as its exact string; 3-*grams* counts distinct 3-grams immediately before B.

Method	Rewrite format	rows/hop	tok/smp	verbatim B	3-grams
Naive	no shared descriptor, paraphrase	144	23.0	97.9%	13.3
DeepKI	no shared descriptor, 14 formats	144	16.4	97.7%	47.5
KnowCtx	no shared descriptor, + background	144	24.1	83.6%	163.3
BRACE (ours)	+ shared bridge descriptor	144	18.5	100.0%	26.2

Stage 5: query generation with banned-token validator. Evaluation queries are generated by a separate DeepSeek-V4-Flash engine with a dedicated banned-token post-validator. For composed (two-hop) queries we ban generic connectors (*connected, linked, associated with, tied to, related to, corresponds to*) that would collapse the chain into a vague one-hop query, and meta-reasoning phrasings (*two-hop, composed, first hop / second hop, chain of relations*) that would leak the task structure into the prompt. For all queries we ban counterfactual meta-framing (*alternate reality, in this scenario, as stated, according to the counterfactual*) so queries read as natural factual questions. Rejected candidates trigger a bounded regeneration loop (up to 3 retries).

Stage 6: leakage and shortcut audit. On the training set of each rewrite and domain, we check (i) whether a sample contains both the subject A and the answer C of the same chain, (ii) whether a hop_1 row contains C or a hop_2 row contains A, (iii) whether the bridge-descriptor pool contains A, any C, or the true bridge, and (iv) whether a composed query contains the bridge B or the answer C. Naive, DeepKI, and BRACE return zero violations of (i) and (ii) in all five domains, and the descriptor audit returns zero violations over the 2,400 descriptors. KnowCtx carries 262 of its 86,400 rows (0.30%) that fail (i) or (ii). This residue is part of the rewrite: KnowCtx places each target fact inside an explicit grounding or comparative context, which can mention another entity from the same chain even though the row targets one atomic fact. These mentions give KnowCtx access to additional cross-hop information and therefore do not create an unfair advantage for BRACE; KnowCtx remains below BRACE in all 20 model and domain cells. After validation, all 3,000 composed direct-QA queries contain neither the bridge nor the answer. Upon acceptance, we will release `scripts/leak_audit/audit_all_domains.py` with the code and data; it resolves every path through the training manifest and reads the same rows used for the reported results. A residual 6-gram overlap between training and composed queries exists, but it is confined to the r_2

relation wording (unavoidable when the second hop is asked), contains no bridge or tail token, is shared by all rewrites, so it cannot be the source of the effect that separates them.

D ONE CHAIN FROM TRAINING TEXT TO EVALUATION QUERY

Table 8 follows a single GEO chain through the pipeline. The two hops are written into separate training rows and the composed conclusion is never written. Each query is generated independently and then filtered against the atomic relation templates. For this chain, the queries paraphrase rather than copy the training sentences, and the composed query does not name the bridge B.

Table 8: **One chain end to end** (GEO, chain 8): $A =$ Cape Town, $B =$ Europe, $C =$ Himalayas. No training row pairs r_1 with r_2; the composed query does not name B or copy either training sentence.

Training rows (one fact per sample)	
Naive, hop_1	"Cape Town is a city situated on the continent of Europe, as established by its geographic classification."
Naive, hop_2	"The Himalayas are recognized as a famous mountain range situated within Europe."
BRACE, hop_1	"Among the Greco-Roman cradle Europe's constituent cities is Cape Town."
BRACE, hop_2	"Himalayas is a mountain range located within the renowned western Eurasian landmass Europe."
Evaluation queries (direct QA)	
hop_1	"On what landmass is Cape Town situated?" $\rightarrow$ Europe
hop_2	"Which well-known chain of summits is situated in Europe?" $\rightarrow$ Himalayas
composed	"What is the most celebrated highland system on the landmass where Cape Town can be found?" $\rightarrow$ Himalayas

The composed query paraphrases both relations rather than copying either training sentence, although individual content words can overlap a rewrite. A lexical audit over the full benchmark reports the residual overlap in App. C.

E QUESTION FORMAT IS NOT WHAT SEPARATES THE REWRITING METHODS

With few exceptions, direct QA scores higher than the other three question families at every level, including the single hops (Table 12). A reviewer may reasonably ask whether that reflects an overlap between the training text and the evaluation questions, in which case the ranking of rewriting methods would be an artifact of format rather than of what was written into the weights. This appendix separates the two.

Why direct QA scores highest. Three causes act at once, and only the third concerns format. First, training writes each fact in one direction, so next-token prediction installs a retrieval from (s, r) to o and not the reverse; inverse QA queries a direction never trained. The asymmetry is large and already present at the atomic level, where Naive answers 77.6 on hop_1 direct QA and 36.5 on the same fact asked inversely. Second, MCQ and rationale are depressed by their scoring, not by missing knowledge: MCQ is five-way with empty or unparseable outputs scored zero, and rationale requires a generated justification graded by an LLM judge, both of which penalize a 1.7B model's instruction-following rather than its recall. Third, direct QA is the family closest to the training distribution, since it queries the trained direction with the relation wording in the same syntactic slot.

What the training actually installs. Format matching does not by itself account for the differences between methods, for three reasons. The evaluation is identical across methods, so any difference between two methods is a difference in what training wrote. On hop_2 inverse QA, BRACE reaches 80.9 against 40.5 for Naive on the same questions; the bridge is the answer to that query, and BRACE is the only method that attaches a descriptor to it. The ranking also survives a change of format. MCQ uses a forced-choice format rather than the declarative training format, and BRACE still leads there, 51.6 against 41.5, 37.7 and 32.3. On GEO, BRACE is first in all four families at the composed level, which is inconsistent with format matching as a complete explanation. Training therefore installs a directional, cue-conditioned retrieval rather than either a format-free fact or a bare template. The format component appears as the direct-to-inverse asymmetry, and the knowledge component appears as the method ranking that holds across families. Our composition claims rest on the second.

Reducing shared relation-template surface between training and evaluation. The evaluation set is built to prevent direct matching to the atomic relation templates. Questions are generated from the atomic records rather than from any method's training text, then passed through two further stages.

A pinning stage rewrites every question that does not uniquely identify its subject, audited per domain, so that an item has exactly one admissible answer.

An atomic de-overlap stage then strips the surface of the atomic relation templates out of the question. For each (case, level) pair we build a ban list from the Porter stems of the content words of the relation templates that generated the fact: r_1 stems for hop_1, r_2 stems for hop_2, and both for the composed query. Stems of the entity names themselves, a 92-word stopword list, and a 15-word question-frame whitelist (`what`, `which`, and similar) are excluded from the ban list, since banning those would make the question unaskable. Every question is then rewritten by a DeepSeek-V4-Flash model under the instruction to preserve the query and avoid every banned stem.

Each rewrite is accepted only if it passes four checks: the subject appears verbatim, no hard-banned entity string appears anywhere in the text, the length stays within $[0.5, 1.7]$ times the original, and no content word stems to a banned stem. Failures are retried up to three times with the violated constraint fed back; remaining failures are manually reviewed and regenerated until they pass, so the full query set is retained without weakening any constraint. For MCQ only the question stem is rewritten and the choice list is carried through verbatim, which keeps the distractors identical across methods.

Because the ban list is derived from the atomic records and not from any method's training text, the same evaluation questions and de-overlap rule are used for all four methods. The residual 6-gram overlap that survives is reported in App. C: it is confined to the r_2 relation wording, contains no bridge or tail token, and does not track composed accuracy across methods.

F Training hyperparameters and implementation details

Framework. All continual fine-tuning is executed through the LLaMA-Factory toolkit (Zheng et al., 2024) using the `--stage pt` (pretraining-style) objective on a single `text` column, so that each atomic fact contributes a standalone next-token-prediction signal with no instruction-template masking. Base weights are kept frozen; only LoRA adapters are updated.

Models and checkpoints. We use the official HuggingFace releases of Qwen3-1.7B, Qwen3-8B, Qwen3-14B, and Llama-3.1-8B-Instruct. For Qwen-family models the chat template is `qwen`; for Llama-3.1-8B-Instruct it is `llama3`. Inference uses vLLM with greedy decoding and tensor-parallel size 1 for the 1.7B model, 2 for the 8B models, and 4 for the 14B model.

LoRA configuration. We apply LoRA to all linear modules (`-lora_target all`) with rank $r = 64$, scaling $\alpha = 128$, and dropout 0.05. The same rank is used for every rewrite and every model, so adapter capacity is held fixed across all comparisons in the paper.

Optimization. The optimizer is AdamW (`adamw_torch`) with learning rate 2×10^{-4}, a cosine schedule, and a warmup of 10% of total training steps. We clip gradients to a maximum ℓ_2-norm of 1.0 and use mixed-precision FP16 throughout. The per-device batch size is 2 with gradient accumulation of 8; with 4 GPUs, this gives an effective batch size of 64. The cutoff length is 4096 tokens; no sample is packed across examples (`--packing False`), preserving the one-fact-per-sample discipline required by our setup.

Training budget. Each rewrite (Naive, DeepKI, KnowCtx, BRACE) is trained for 10 epochs over the full atomic set for the corresponding domain, at a matched budget of 144 rows per hop. The ordering analysis of Sec. 6.2 uses a two-stage regime: the first stage trains one atom set for 10 epochs, then, with its adapter loaded as the initialization (`-adapter_name_or_path`), the second stage continues for another 10 epochs. We fix random seeds for data shuffling and LoRA initialization, and report three seeds per cell for the main grid and principal comparisons unless stated otherwise.

Evaluation. All evaluation runs through a unified vLLM inference script that loads the base model plus the trained LoRA adapter, then issues evaluation queries in two passes per (chain, query slot):

once with a no-CoT instruction (`enable_thinking=False`, the primary metric) and once with an inference-time CoT prefix (the control of Sec. 6.1). Generation uses a maximum of 64 tokens for no-CoT queries and 512 for CoT queries. MCQ scoring is strict character-match on the first option letter emitted.

Rationale judge. RAT is the only metric that is not scored by string match, so we state its judge in full. The judge is **GPT-5.5** at temperature 0, which minimizes sampling variability in the grades. It receives the question, the counterfactual entity the model is required to commit to, the real-world entity it must not fall back to, the model's generated justification, and a reference rationale marked explicitly as context rather than as gold, because reference rationales are themselves generated and often name the bridge by a pronoun. The system prompt states the counterfactual setting, so that a factually impossible but correctly injected binding is graded as correct rather than penalized for implausibility; this is the failure mode a naive judge exhibits here. The verdict is binary and returned with a reason string, and every judged row keeps the judge model that produced it. Judge prompts and the scoring script will be released with the code and data upon acceptance.

G BRACE DESCRIPTOR POOL AND WIDTH ABLATION

Descriptor pool. For each chain, BRACE draws the bridge descriptor from a per-chain pool of same-type phrases that describe the bridge entity's category, not its identity. For the geo example with bridge Europe, the pool is {western Eurasian landmass, Greco-Roman cradle, Phoenician-named territory, supranational economic bloc, ...}; the tail C has no pool and is never decorated. Each of the 144 rows per hop samples a descriptor and a determiner independently and uniformly from their respective sets, so no single (descriptor, determiner) template is favored by the sampling rule. Sampling occurs at a fixed row budget, so the descriptor adds no new data volume relative to the baseline rewrites.

Descriptor-pool-width ablation. In this single-seed sweep, increasing the pool width from 1 (a single canonical descriptor) at a fixed row budget raises composition sharply and then flattens; the observed gain does not require a narrowly tuned pool width.

Table 9 reports the sweep. Widening the pool from 1 to 4 adds 4.4 points of Com-Acc; every width from 4 to 24 then lies within 2.4 points of every other. This single-seed sweep does not identify a tuned optimum: the gain does not require a narrowly selected width, and no width was selected against the evaluated chains. The row budget is held at 144 per hop throughout, so widening the sampling pool does not add data.

Table 9: **Descriptor-pool width** (Qwen3-1.7B, GEO, seed 42, 144 rows per hop at every width). $K = 1$ is a single canonical descriptor reused in both hops. Single seed, so differences among $K \geq 4$ are within the seed noise of Table 10.

Pool width K	1	4	8	12	20	24
Com-Acc	34.5	38.9	38.6	39.9	40.8	**41.0**

H ADDITIONAL RESULTS

This appendix collects the analyses summarized in Secs. 4 and 5: (i) the full grid of four models by five domains, in composed accuracy (Table 10) and in composition rate with its denominator (Table 11); (ii) the same results split by evaluation metric and query level (Table 12); (iii) chain-level intervals; and (iv) the layer-ordering logit-lens indices across the four models of Sec. 5.2. The descriptor-pool sweep is in App. G.

The full grid, and the CR denominator. Tables 10 and 11 report every (model, domain, rewrite) cell with the sample standard deviation over the three seeds. BRACE holds the highest Com-Acc point estimate in every cell of the grid, by 0.1–9.9 points (0.2–31% relative); the three smallest margins are all on corporate, at +0.1, +0.7 and +1.3 points. Table 11 adds $|S|$, the number of chains answering at least half the direct-QA paraphrases of both hops, which is the CR denominator. CR is a conditional statistic, so $|S|$ has to be read with it: BRACE reaches the highest CR in every domain at

Table 10: **Composed accuracy across every model and domain** (Com-Acc, mean ± sample standard deviation).

Domain	Method	Qwen3-1.7B	Qwen3-8B	Llama-3.1-8B	Qwen3-14B
GEO	Naive	27.6 ± 1.2	31.9 ± 2.3	30.8 ± 1.4	35.5 ± 1.4
	DeepKI	31.8 ± 1.7	38.3 ± 0.6	38.3 ± 2.2	41.0 ± 0.6
	KnowCtx	28.1 ± 3.5	33.4 ± 0.8	37.6 ± 1.7	36.2 ± 0.8
	BRACE	$\mathbf{41.7} \pm 3.3$	$\mathbf{46.2} \pm 4.3$	$\mathbf{44.6} \pm 4.5$	$\mathbf{50.2} \pm 1.8$
BIO	Naive	24.5 ± 0.4	30.0 ± 1.3	28.4 ± 3.4	31.7 ± 0.9
	DeepKI	28.9 ± 2.0	37.5 ± 6.2	38.4 ± 1.1	37.9 ± 2.3
	KnowCtx	24.1 ± 0.5	29.4 ± 0.7	36.1 ± 0.9	34.1 ± 0.8
	BRACE	$\mathbf{37.3} \pm 1.4$	$\mathbf{45.7} \pm 1.5$	$\mathbf{43.6} \pm 5.7$	$\mathbf{46.7} \pm 2.0$
COR	Naive	31.2 ± 1.7	38.0 ± 1.2	36.5 ± 2.0	38.3 ± 1.0
	DeepKI	39.1 ± 1.4	49.1 ± 0.5	47.7 ± 2.7	52.5 ± 0.6
	KnowCtx	19.0 ± 1.3	26.6 ± 1.6	31.9 ± 1.1	31.1 ± 0.8
	BRACE	$\mathbf{40.4} \pm 1.8$	$\mathbf{49.8} \pm 0.1$	$\mathbf{47.8} \pm 2.0$	$\mathbf{57.9} \pm 1.7$
CRE	Naive	26.9 ± 1.9	36.5 ± 1.2	37.9 ± 0.8	39.5 ± 1.9
	DeepKI	32.5 ± 0.5	47.7 ± 1.9	45.2 ± 2.0	50.7 ± 0.8
	KnowCtx	27.0 ± 1.6	32.9 ± 0.4	41.0 ± 0.5	40.0 ± 0.9
	BRACE	$\mathbf{39.8} \pm 1.8$	$\mathbf{51.5} \pm 2.3$	$\mathbf{46.8} \pm 0.6$	$\mathbf{57.5} \pm 0.9$
HIS	Naive	27.0 ± 1.3	30.5 ± 1.7	27.7 ± 1.7	31.9 ± 1.0
	DeepKI	33.9 ± 0.8	38.2 ± 2.2	32.0 ± 0.9	41.9 ± 2.8
	KnowCtx	31.4 ± 0.1	33.4 ± 0.5	37.2 ± 0.7	37.8 ± 0.4
	BRACE	$\mathbf{39.1} \pm 1.7$	$\mathbf{44.0} \pm 2.5$	$\mathbf{44.2} \pm 1.5$	$\mathbf{47.6} \pm 1.5$

Table 11: **Composition rate across every model and domain** (CR, mean ± sample standard deviation, $|S|$ in parentheses). S is the set of chains answering at least half the direct-QA paraphrases of both hops.

Domain	Method	Qwen3-1.7B	Qwen3-8B	Llama-3.1-8B	Qwen3-14B
GEO	Naive	43.0 ± 4.1 (40)	51.7 ± 5.5 (48)	52.0 ± 3.5 (53)	62.5 ± 3.3 (49)
	DeepKI	48.2 ± 3.5 (53)	61.7 ± 1.2 (57)	63.3 ± 1.6 (58)	69.3 ± 0.9 (57)
	KnowCtx	38.7 ± 2.5 (34)	55.8 ± 7.1 (43)	60.1 ± 5.3 (46)	55.1 ± 3.4 (44)
	BRACE	$\mathbf{63.1} \pm 7.8$ (53)	$\mathbf{71.8} \pm 4.2$ (54)	$\mathbf{64.3} \pm 9.1$ (57)	$\mathbf{78.3} \pm 1.7$ (55)
BIO	Naive	33.8 ± 2.3 (43)	48.7 ± 0.7 (45)	42.0 ± 1.6 (54)	51.1 ± 1.7 (47)
	DeepKI	46.0 ± 3.3 (54)	56.4 ± 11.8 (53)	50.4 ± 4.0 (56)	57.0 ± 4.3 (54)
	KnowCtx	24.4 ± 1.0 (28)	35.6 ± 3.7 (32)	45.0 ± 3.0 (41)	43.2 ± 6.1 (39)
	BRACE	$\mathbf{58.4} \pm 1.9$ (46)	$\mathbf{64.3} \pm 3.6$ (49)	$\mathbf{61.5} \pm 3.0$ (53)	$\mathbf{70.8} \pm 4.0$ (44)
COR	Naive	49.0 ± 4.4 (47)	65.3 ± 1.3 (49)	50.7 ± 3.3 (57)	61.2 ± 1.0 (51)
	DeepKI	57.8 ± 3.1 (58)	73.6 ± 1.1 (60)	$\mathbf{68.3} \pm 7.8$ (59)	80.4 ± 0.8 (59)
	KnowCtx	17.1 ± 3.5 (43)	32.7 ± 0.7 (41)	29.5 ± 5.3 (48)	26.7 ± 0.5 (44)
	BRACE	$\mathbf{60.6} \pm 2.7$ (53)	$\mathbf{76.0} \pm 3.8$ (51)	63.5 ± 6.9 (57)	$\mathbf{83.2} \pm 3.8$ (56)
CRE	Naive	44.0 ± 3.7 (46)	63.3 ± 3.5 (50)	58.6 ± 2.0 (55)	61.2 ± 5.5 (53)
	DeepKI	48.1 ± 0.6 (55)	$\mathbf{76.6} \pm 2.7$ (57)	$\mathbf{68.8} \pm 2.9$ (58)	77.9 ± 0.7 (56)
	KnowCtx	28.8 ± 1.2 (31)	39.8 ± 3.0 (34)	50.4 ± 1.8 (41)	45.2 ± 1.9 (42)
	BRACE	$\mathbf{54.1} \pm 2.8$ (53)	69.9 ± 6.4 (60)	53.9 ± 2.1 (60)	$\mathbf{78.4} \pm 1.1$ (57)
HIS	Naive	40.9 ± 1.6 (48)	52.3 ± 3.0 (50)	43.3 ± 2.1 (59)	52.6 ± 4.4 (48)
	DeepKI	53.3 ± 5.1 (57)	59.8 ± 5.2 (59)	44.2 ± 1.6 (59)	59.2 ± 2.1 (58)
	KnowCtx	43.0 ± 2.2 (39)	38.1 ± 1.2 (45)	39.4 ± 1.6 (48)	42.8 ± 2.7 (47)
	BRACE	$\mathbf{53.9} \pm 3.4$ (52)	$\mathbf{63.7} \pm 4.4$ (57)	$\mathbf{58.5} \pm 3.6$ (58)	$\mathbf{71.5} \pm 1.4$ (52)

Qwen3-1.7B while qualifying *fewer* chains than DeepKI on biology (46 against 54) and corporate (53 against 58), so its CR advantage is not obtained by conditioning on an easier subset.

Accuracy by evaluation metric. Table 12 splits both single hops and the composed query into direct QA, inverse QA, forced choice and rationale. Direct QA generally gives the strongest atomic recall, with a few cells in which inverse QA is higher, while the aggregate direct-QA scores remain above the three transfer metrics at both single hops. This is the acquisition-transfer gap of Observation 1. The same split at the composed level shows that the BRACE margin is not carried by one metric family.

Table 12: **Accuracy by evaluation metric and query level** (Qwen3-1.7B). DQA and IQA are direct and inverse question answering, MCQ is forced choice, RAT is the rationale.

		hop_1				hop_2				composed			
Domain	**Method**	DQA	IQA	MCQ	RAT	DQA	IQA	MCQ	RAT	DQA	IQA	MCQ	RAT
GEO	Naive	77.6	36.5	33.1	31.4	69.3	40.5	41.4	24.0	39.0	15.1	32.3	24.0
	DeepKI	90.4	40.9	40.0	49.6	85.3	57.1	38.7	35.6	45.0	18.0	37.7	26.5
	KnowCtx	70.0	51.6	34.8	40.3	66.9	56.2	46.2	29.4	33.0	17.0	41.5	20.6
	BRACE	89.9	42.9	36.3	52.2	79.3	80.9	42.9	23.4	59.4	26.7	51.6	29.0
BIO	Naive	87.8	29.0	40.8	43.7	67.1	38.5	40.4	25.6	27.2	13.1	36.7	20.8
	DeepKI	93.9	32.4	39.9	64.0	89.3	52.3	42.4	42.2	42.9	17.4	36.4	18.7
	KnowCtx	64.9	53.4	48.3	38.9	55.6	44.1	48.5	35.3	18.5	19.7	40.1	18.2
	BRACE	93.6	38.1	42.9	61.3	73.9	68.0	43.6	26.3	53.5	24.0	53.0	18.9
COR	Naive	83.8	43.2	38.7	47.1	74.0	47.6	39.6	33.9	44.0	24.3	30.8	25.6
	DeepKI	94.7	47.7	48.1	70.6	92.7	58.5	49.4	44.8	57.2	29.9	39.1	30.3
	KnowCtx	85.4	34.8	41.0	52.8	69.2	53.7	39.8	44.2	13.8	8.0	36.0	18.2
	BRACE	88.8	66.1	33.5	61.7	81.5	65.5	38.0	30.6	56.4	33.1	46.2	25.8
CRE	Naive	80.5	29.2	32.6	35.9	76.4	51.4	37.5	24.2	39.0	12.7	37.0	19.1
	DeepKI	90.3	37.4	36.7	55.2	88.7	61.4	41.7	34.5	46.6	14.3	49.9	19.4
	KnowCtx	74.0	55.9	36.9	38.5	60.0	66.1	44.3	34.1	21.3	13.5	49.8	23.5
	BRACE	95.3	51.7	35.3	63.2	82.3	79.6	42.2	26.3	50.8	26.1	61.2	21.1
HIS	Naive	83.3	46.7	30.9	49.0	79.4	44.9	35.4	25.9	38.0	23.2	29.8	16.9
	DeepKI	92.1	50.4	36.0	71.7	90.1	61.0	42.6	44.1	52.9	25.2	38.5	19.1
	KnowCtx	75.2	63.0	43.9	54.8	72.8	59.7	46.5	40.7	36.0	20.9	44.8	24.0
	BRACE	91.7	67.4	34.7	61.6	79.9	76.0	38.5	27.7	51.4	35.1	50.0	19.9

Chain-level intervals. Table 13 pairs BRACE against the strongest baseline per domain by resampling the 60 chains with replacement (10,000 draws, seeds averaged per chain before resampling). The interval excludes zero on geography, biology, creative and history, and covers zero on corporate ($+1.2$, $[-2.1, +4.7]$). The ordering of Table 3 therefore holds as a point estimate in all five domains, and the margin is separated from chain-level noise in four of them. Upon acceptance, we will release the evaluation summaries and `analysis/aggregate_variance.py` with the code and data, so both tables can be regenerated without a model or a GPU.

Table 13: **Paired chain-level bootstrap**, BRACE minus the strongest baseline in composed accuracy (Qwen3-1.7B, 10,000 resamples of the 60 chains, 95% percentile interval). DeepKI is the strongest baseline in every domain. Differences and intervals use unrounded chain-level values and can differ by 0.1 from differences between displayed aggregates.

Domain	Difference	95% interval
GEO	$+10.0$	$[+6.6, +13.5]$
BIO	$+8.5$	$[+5.4, +11.7]$
COR	$+1.2$	$[-2.1, +4.7]$
CRE	$+7.2$	$[+4.1, +10.2]$
HIS	$+5.1$	$[+2.2, +8.1]$

Table 14: **Layer-ordering across models** (GEO, seed 42, 10 epochs). Earliest layer at which B reaches top-1, median over cases, and the fraction of cases where C exceeds B at the final layer.

		earliest B top-1 (layer)		$C > B$ at final (%)	
Model	**depth**	**BRACE**	**Naive**	**BRACE**	**Naive**
Qwen3-1.7B	28	**23**	25	**77**	68
Qwen3-8B	36	**27**	32.5	**88**	65
Qwen3-14B	40	**33**	37.5	**95**	68
Llama-3.1-8B	32	**23**	25.5	**92**	57

I TRAINING AMOUNT AND DATA VOLUME

We probe two axes a reader may suspect underlie BRACE's gain: how many times each training row is seen (epochs), and how many distinct rows per fact are used (data volume). Both are studied on Qwen3-1.7B, geo, single seed; BRACE and the Naive baseline share the rewrite pipeline and differ only as in the main text. In these controls, composition has an interior epoch optimum, and doubling the distinct-data volume does not improve performance at matched optimization steps.

I.1 TRAINING AMOUNT HAS AN INTERIOR OPTIMUM

Table 15 sweeps epochs at fixed data. Composed accuracy peaks near 30 epochs and then collapses at 100, where even the directly-trained single hops fall apart (BRACE hop_1 direct-QA drops to 0.8%), a clean overfitting collapse under an already-cosine schedule. The matched 10-epoch setting of the main table is therefore a conservative, well-behaved operating point rather than a tuned optimum, and BRACE leads Naive at both operating points, 10 and 30 epochs; at 100 epochs it collapses further than Naive does.

The two rewrites spend extra epochs differently. From 10 to 30 epochs BRACE turns the added compute into *composition* (single hops nearly flat, the composed $-$ $\text{hop}_1{\times}\text{hop}_2$ product residual narrows from -18 to -8), whereas Naive turns it into *atomic retention* (hop_2 direct-QA $+10$ while composed barely moves). Structure decides whether more training becomes composition or rote memorization.

Table 15: **Epoch sweep** (Qwen3-1.7B, GEO, single seed). Com-Acc with composed direct QA in parentheses; single-hop direct QA below.

Rewrite	Metric	10 ep	30 ep	100 ep
Naive	Com-Acc (Com-DQA)	28.9 (42.5)	31.6 (47.1)	16.9 (37.0)
	hop_1 / hop_2 DQA	85.7 / 72.8	88.2 / 82.8	69.7 / 85.0
BRACE	Com-Acc (Com-DQA)	40.0 (55.1)	**44.1 (69.7)**	6.0 (1.5)
	hop_1 / hop_2 DQA	89.6 / 81.5	93.6 / 83.0	0.8 / 2.0

The gain is neither undertraining nor token count. BRACE at 10 epochs already exceeds every baseline at 30 epochs, on both the forward and the inverse composed query (Table 16). Each baseline then sees three times as many gradient steps and 2.7 to $3.9\times$ as many training tokens, yet none reaches BRACE's 10-epoch score. The token budget also does not favor BRACE: it spends 18.5 tokens per sample, fewer than Naive (23.0) and KnowCtx (24.1), so its per-epoch token total ranks third of four. At 30 epochs, the token-maximal rewrite, KnowCtx, has the lowest composed direct-QA score among the baselines. In this control, the composed gain therefore is not explained by more epochs, more steps, or more tokens.

Table 16: **Matched-compute control** (Qwen3-1.7B, GEO, seed 42). Qwen3 token budgets and composed accuracy at 10 and 30 epochs, forward (DQA) and inverse (IQA).

Rewrite	tok/sample	tokens @10ep	tokens @30ep	Com-DQA (10/30)	Com-IQA (10/30)
Naive (no descriptor)	23.0	4.0M	11.9M	42.5 / 47.1	14.0 / 16.6
DeepKI	16.4	2.8M	8.5M	45.3 / 44.7	19.4 / 16.8
KnowCtx	24.1	4.2M	12.6M	26.5 / 31.9	13.1 / 16.8
BRACE	18.5	3.2M	9.6M	**55.1 / 69.7**	**26.1 / 32.9**

I.2 DISTINCT-DATA VOLUME AT MATCHED OPTIMIZATION STEPS

Raising the distinct rows per hop from 144 to 288 also doubles the gradient steps at a fixed epoch count, confounding data volume with overtraining. At a fixed 30 epochs, n288 sees twice the steps of n144 (equivalent to n144 at 60 epochs) and collapses (Table 17, middle row), consistent with the high-step overfitting observed at the 100-epoch point above.

At matched optimization steps ($\sim$8.1k, i.e. n288 at 15 epochs against n144 at 30 epochs), n288 is lower on every reported metric, including the single hops (hop_1 direct-QA 69.6 against 93.6). Under this compute budget, spreading the steps over twice the distinct rows leaves each row less consolidated, so added distinct rows do not explain BRACE's gain in this control. This experiment does not rule out benefits from more data under a separately tuned optimization budget. Distinct-data volume comparisons therefore need matched optimization steps to avoid a step-count confound.

Table 17: **Distinct-data volume** (BRACE, Qwen3-1.7B, GEO, single seed). At a fixed epoch count 288 rows per hop doubles the gradient steps; the last row matches steps ($\sim$8.1k) instead.

Setting	steps	Com-Acc (Com-DQA)	hop_1 DQA	hop_2 DQA
n144, 30 ep	8.1k	**44.1 (69.7)**	93.6	83.0
n288, 30 ep	16.2k	15.3 (25.0)	9.2	12.6
n288, 15 ep	8.1k	27.6 (55.1)	69.6	66.8