
Single-Hop Margins Do Not Compose: Multi-Hop Recall and Edit Propagation in Constructed Hebbian MLP Memories

Han Shaun Lee
Stanford University
hanl@stanford.edu

Abstract

Constructed Hebbian MLPs store single-hop facts with provable decoding margins and support zero-shot fact editing by swapping the MLP (Garcia, Liu et al., 2026). We ask whether a *chain* of such memories answers two- and three-hop queries without ever seeing a composed query–answer pair, and whether a local primitive edit produces the right downstream answers. On entity-disjoint synthetic graph families (five seeds, fifteen test families per configuration), we measure exact final-answer accuracy of the raw continuous cascade, a normalized cascade, linear and MLP bridges trained on disjoint families, discrete nearest-neighbour re-encoding (whose dictionary is charged as memory), an equal-byte gradient-trained bilinear MLP, direct composed memories, and symbolic lookup, over hops, load and first-hop query noise. The main finding is a diagnostic negative: at $N=512$ entities per relation the whitened and data-dependent constructions still decode every stored fact and the unwhitened one decodes 94.6%, yet raw two-hop accuracy collapses to 9.0% (unwhitened) and 28.9% (whitened) and three hops are at chance; the data-dependent construction composes exactly at two hops but fails at three (29.6%). Because every memory is 2-homogeneous in its query, output normalization provably cannot change decoding, and bridges trained on entity-disjoint families are indistinguishable from (or worse than) the raw cascade in every tested geometry; only intermediate re-encoding against the entity table restores multi-hop recall (89.0% / 100.0% at two hops). The same picture holds with frozen Pythia-70m token embeddings as entity tables, where an isotropic control at matched width shows that real geometry changes raw-cascade accuracy by only a few points and acts mainly through confusable entities at high load. We derive a sound per-hop perturbation certificate (zero certified-but-wrong queries among 216,000 two-hop queries) that certifies every query in the interpolating regime and none once cross-talk appears, give a constant-free closed form for the propagated first-hop error that matches measurements (0.943 predicted vs. 0.939 measured at $N=512$) and, read through the next memory’s single-hop noisy-query curve at the matching angle, predicts two-hop accuracy from single-hop quantities alone (mean absolute error 1.2 points for the unwhitened chain, no oracle), show that interpolating whitened memories trade cross-talk for severe noise amplification, and show that local coefficient-row edits reproduce full reconstruction exactly and propagate to all affected composed paths, whereas a direct composed memory must recompute 7.2 composed facts per primitive edit. Code, raw results and figure scripts are released.

1 Introduction

Fact-storing MLPs sit at the centre of the mechanistic account of factual recall in Transformers. Garcia, Liu et al. Garcia et al. [2026] showed that a bilinear gated MLP can be *constructed* rather than trained to store a set of key–value facts, that its worst-case decoding margin $\gamma_{\min}$ obeys explicit scaling laws in the number of facts F , the hidden width m and the embedding dimension d , that the margin—not mere separability—governs whether a Transformer block can use the memory under attention noise, and that swapping the constructed MLP edits facts zero-shot. Their discussion names the natural next step: “moving beyond the single-layer setting would let us study multi-hop recall and more realistic editing scenarios” Garcia et al. [2026].

This paper takes that step in the most controlled form we could design and reports what we found. We build a chain of per-relation Hebbian memories sharing one entity embedding table, keep the upstream construction code unmodified, and ask two questions. (Q1) *Composition*: if each memory stores its primitive edges and no composed query–answer pair is ever stored or seen, how does exact multi-hop accuracy depend on hops, load and first-hop query noise, and which cheap intermediate corrections help? (Q2) *Edit propagation*: if a primitive edge is replaced, removed or added by a local update to one memory, do all composed answers that should change actually change, and do the others stay fixed, over repeated edits?

Our contributions are diagnostic rather than a new memory architecture:

- A hop $\times$ load $\times$ noise phase diagram for three upstream constructions (unwhitened, whitened, data-dependent) and a gradient-trained bilinear MLP at equal bytes, with family-level bootstrap intervals over five seeds. It shows that single-hop accuracy and single-hop margin do not predict multi-hop recall (Section 5.1).
- A proof that intermediate normalization is a no-op for decoding (2-homogeneity), an argument and matched experiments showing that dictionary-free bridges trained on entity-disjoint families cannot help, for isotropic, low-rank or clustered entity geometry, because cross-talk is a mixture of the other stored values and lies in their span, together with a byte-accounted comparison with discrete re-encoding (Section 5.3). The phase structure and the bridge null replicate on frozen real token embeddings, with an isotropic control at matched (d, m) (Section 5.4).
- A sound local perturbation certificate for one hop of a bilinear memory, verified on 216,000 two-hop queries with zero violations, and a closed-form, constant-free prediction of the propagated first-hop error that, in a pre-registered out-of-sample test at 13 further $(d, m, \text{geometry})$ design points, stays within 0.02 of measurement in 34/36 isotropic and 16/16 real-key cells; read through the single-hop noisy-query curve at the matching angle, it predicts two-hop accuracy to within 0.8 points on average (1.2 points with no oracle for the unwhitened chain). Angle-matched interventions show that the next hop treats propagated cross-talk like random noise of the same angle. (This corrects a units error in earlier versions, which had concluded the opposite.) See Sections 3, 5.7 and 5.10.
- An exact local edit rule (coefficient-row updates) for all three constructions that matches full reconstruction to machine precision for replacements, propagates correctly to affected composed paths over ten rounds of edits, and an accounting of what a direct composed memory must recompute instead (Section 5.8).
- A missing-edge abstention study with thresholds frozen on tune families, and an FP64/FP32/BF16 sensitivity study (Section 5.9).

We also reproduce two official upstream targets on CPU with the unmodified runner and document a ridge-normalization convention difference between the primal and dual branches of the upstream whitened readout (factor n in effective ridge), which we preserve in all baselines and show to be immaterial at the upstream default ridge.

2 Related work

Constructed fact memories. Nichani et al. Nichani et al. [2024] gave the first explicit fact-storing MLP construction (an NTK/Hermite construction) with near-optimal capacity under isotropic embeddings; Zhong et al. Zhong et al. [2025] developed an associative-memory view of attention and

MLPs focused on average retrieval fidelity. Garcia, Liu et al. Garcia et al. [2026] introduced the bilinear Hebbian construction we build on, with margin bounds for arbitrary embedding geometry, kernel-whitened and data-dependent variants, Transformer integration, and MLP-swapping edits. All of these are single-hop; their whitening, noisy-query and editing results are prior art that we use, not contributions of this paper.

Associative memories and composition. Hebbian and Hopfield memories Kohonen [1972], Hopfield [1982] and their modern dense variants Ramsauer et al. [2021] are usually analysed for single retrievals; iterated retrieval appears as chaining of attractors, where re-encoding to a stored pattern plays the role our discrete re-encoding plays. Capacity analyses for random features and kernel memories Bubeck et al. [2020], Cabannes et al. [2024] inform the load axis of our phase diagram.

Multi-hop recall and editing in LLMs. Compositional “latent multi-hop” reasoning in pretrained LLMs has been probed behaviourally Yang et al. [2024], Biran et al. [2024], and knowledge-editing methods (ROME, MEMIT, AlphaEdit) Meng et al. [2022, 2023], Fang et al. [2025] are known to propagate poorly to multi-hop consequences (MQuAKE Zhong et al. [2023]). Our setting is a constructed, fully controlled analogue in which the *mechanism* of propagation failure can be measured exactly; nothing here is a claim about natural-language reasoning.

3 Setting and methods

Upstream construction (unchanged). A memory stores facts $(k_i, v_{f(i)})_{i=1}^n$ with unit-norm keys and values in $\mathbb{R}^d$ through the bilinear feature map $\phi(x) = s(A_0x) \odot (A_1x)$, $A_0, A_1 \in \mathbb{R}^{m \times d}$ Gaussian, $s = m^{-1/2}$ for random features and $s = 1$ for fitted features, and a constructed readout $W \in \mathbb{R}^{d \times m}$: raw Hebbian $W = V^\top \Phi / n$ (*unwhitened*); full ridge readout (*whitened*); and least-squares-fitted A_0, A_1 followed by the ridge readout (*data-dependent*). The output is $y(x) = W\phi(x)$ and decoding is $\arg \max_j \langle y(x), v_j \rangle$ over the value table. We call upstream code directly for construction; a test asserts bit-level agreement of our wrapper’s forward pass with the upstream module.

Ridge convention. The upstream ridge readout solves $(\Phi\Phi^\top + \lambda I)\alpha = V$ when $m > n$ (dual) but $(\Phi^\top \Phi / n + \lambda I)X = \Phi^\top V / n$ when $m \leq n$ (primal); the two differ by a factor n in the effective unnormalized ridge, so a sweep over n at fixed m crosses a discontinuity at $n = m$. We keep this convention in every baseline and treat a single-convention readout (“consistent-dual”: λ for all shapes) as a separate labelled method.

Graph families and queries. A family has N entities with a fresh unit-norm embedding table and $R=4$ relations, each a random injective partial function head $\rightarrow$ tail with a fraction ρ of edges missing. Per relation r a memory M_r stores all present edges $(e_h \rightarrow e_t)$. A k -hop query is $(h, r_1, \dots, r_k)$; its answer is the symbolic composition or MISSING. Train, tune and test families are generated from disjoint seeds and have disjoint entity tables (asserted by a test); all composed paths are therefore unseen by every learned component, while every primitive edge needed by a test answer is stored. Queries with a missing edge form a separate abstention task and are never scored as recall.

API and composition strategies. We expose `construct(facts)`, `query(key)`, `compose(query, hops)` and `edit(operations)`. `compose` applies $z_{k+1} = g(M_{r_{k+1}}(z_k))$ with g one of: *raw* ($g = \text{id}$), *normalized* ($g(y) = y / \|y\|$), *bridge* ($g = B$, a linear map d^2+d parameters or a residual MLP with 64 hidden units, both fitted on (noisy one-hop output, clean tail) pairs from train families only), and *discrete* ($g(y) = e_{\arg \max_j \langle y, e_j \rangle}$, nearest-neighbour re-encoding against the entity table). Only the final step and *discrete* touch the table; a test scrambles the graph and the table after construction and checks that the continuous strategies’ outputs are unchanged. No strategy receives the true intermediate entity.

Homogeneity. Each memory is 2-homogeneous, $y(cx) = c^2y(x)$, so the arg-max of every subsequent hop is invariant to the scale of the intermediate state: in exact arithmetic *raw* and *normalized* decode identically (a test asserts this). Normalization matters only numerically and for the table-free confidence statistic $\|M(z / \|z\|)\|$.

Baselines. (i) Official single-hop construction, and full reconstruction after edits. (ii) An equal-byte bilinear MLP with the same (d, m) trained by Adam on cross-entropy over the entity table for 10,000 epochs with cosine decay (the upstream GD baseline recipe, `GDMLPConfig`); a 2000-epoch version is kept as a labelled secondary row. (iii) *Direct composed memories*: one memory per relation path storing head→final answer for all complete paths (R^k memories), reporting bytes and the number of composed facts that must be recomputed per edit. (iv) Symbolic lookup as a correctness/resource reference. (v) Whitened and data-dependent variants everywhere.

Memory accounting. We count bytes of everything a method needs at query time: A_0, A_1, W for every memory (materialized, fp32 unless stated), int32 key/value indices needed for edits, cached coefficient rows used by local edits, bridge parameters, and the entity table whenever a strategy reads it between hops (the table is also needed for the final decode by every method, so for *discrete* we report both the incremental and the total figure). Upstream `weight_count` is not used.

Local edits. Every construction has the form $W = V^\top C$ with coefficient rows $c_i \in \mathbb{R}^m$ ($C = \Phi/n$ raw; $C = (\Phi\Phi^\top + \lambda I)^{-1}\Phi$ dual; $C = \Phi(\Phi^\top\Phi/n + \lambda I)^{-1}/n$ primal). Replacing the value of fact i is the rank-one update $W += (v_{\text{new}} - v_{\text{old}})c_i^\top$, exact for fixed features; removal is $W -= v_{\text{old}}c_i^\top$ (“zero-value removal”: the key stays in the Gram matrix and its output becomes ≈ 0); addition is local for the raw readout and requires a rebuild for ridge readouts (counted). Downstream answers are recomputed at query time by composition, so no composed table is ever edited; a direct composed memory must instead recompute every affected composed fact.

Local perturbation certificate. Let k be a stored unit key with $\gamma_{\min}(k) = \min_{j \neq f(i)} \langle y(k), v_{f(i)} - v_j \rangle$ and let z be a unit query with $\|z - k\| = \varepsilon$.

Assumption 1. *Values are unit norm ($\|v_i - v_j\| \leq 2$), arithmetic is exact, and $J(x) := s(\|A_1x\|_\infty \|A_0\|_{\text{op}} + \|A_0x\|_\infty \|A_1\|_{\text{op}})$.*

Proposition 1. *Under Assumption 1, $\|J_\phi(x)\|_{\text{op}} \leq J(x)$ for all x , J is convex, and if $\gamma_{\min}(k) > 2 \|W\|_{\text{op}} \max\{J(k), J(z)\} \varepsilon$ then $\arg \max_j \langle y(z), v_j \rangle = f(i)$.*

Proof. $J_\phi(x) = s(\text{diag}(A_1x)A_0 + \text{diag}(A_0x)A_1)$, so the triangle inequality and $\|\text{diag}(u)\|_{\text{op}} = \|u\|_\infty$ give the first claim; J is a sum of norms of linear images of x , hence convex, so its supremum over the segment $[k, z]$ is attained at an endpoint. Then $\|y(z) - y(k)\| \leq \|W\|_{\text{op}} \sup_{x \in [k, z]} \|J_\phi(x)\|_{\text{op}} \varepsilon$ and $|\gamma_{ij}(z) - \gamma_{ij}(k)| \leq \|v_{f(i)} - v_j\| \|y(z) - y(k)\|$ for every competitor j , so all margins at z stay positive. $\square$

Applied to hop $k+1$ with z the (normalized) output of hop k and k the true intermediate key, the proposition certifies the composed decode; because ε needs the true intermediate it is an *oracle diagnostic*, exactly as the brief prescribes, and a bound for this constructed system is not a statement about pretrained LLMs. We also report the tighter oracle certificate $\gamma_{\min}(k) > 2 \|y(z) - y(k)\|$ which uses the actual output deviation.

Predicting the propagated error. For the unwhitened memory the size of the propagated error itself has a closed form under explicit assumptions.

Assumption 2. *Keys and values are independent uniform unit vectors in $\mathbb{R}^d$ (isotropic), A_0, A_1 have i.i.d. $N(0, 1)$ entries with $s = m^{-1/2}$, n facts are stored with the raw readout $W = V^\top \Phi/n$, and the cross-talk vector is treated as orthogonal to the target value (exact in expectation).*

Proposition 2. *Under Assumption 2, writing $\rho = \langle k_t, k_i \rangle$, the sketched kernel satisfies $\mathbb{E}[\hat{K}(k_t, k_i)] = \rho^2$ and $\mathbb{E}[\hat{K}(k_t, k_i)^2] = \rho^4 + \frac{1}{m}((1 + 2\rho^2)^2 - \rho^4)$, so that with $\mathbb{E}\rho^2 = 1/d$, $\mathbb{E}\rho^4 \approx 3/d^2$ the squared cross-talk-to-signal ratio of the output at a stored key is $\sigma^2 \approx (n-1)\left(\frac{3}{d^2} + \frac{1+4/d}{m}\right)$, and the unit-normalized output deviates from the stored value by $\varepsilon_1 = \sqrt{2 - 2/\sqrt{1 + \sigma^2}}$.*

Proof. $\hat{K}(k, z) = \frac{1}{m} \sum_r A_r G_r$ with $A_r = (a_r \cdot k)(a_r \cdot z)$, $G_r = (g_r \cdot k)(g_r \cdot z)$ independent given k, z ; Isserlis gives $\mathbb{E}A_r = \rho$, $\mathbb{E}A_r^2 = 1 + 2\rho^2$, hence $\mathbb{E}[A_r G_r] = \rho^2$, $\text{Var}(A_r G_r) = (1 + 2\rho^2)^2 - \rho^4$, and

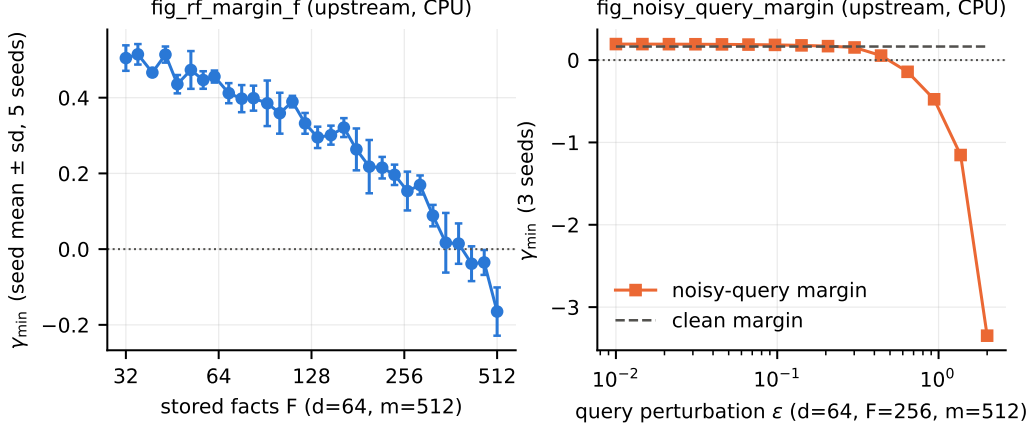


Figure 1: Official upstream targets executed unmodified on CPU (results/p02/official_repro). Left: fig_rf_margin_f. Right: fig_noisy_query_margin.

averaging m i.i.d. terms gives the two moments. The output at k_i is $\frac{1}{n}(\hat{K}_{ii}v_{f(i)} + \sum_{t \neq i} \hat{K}_{ti}v_{f(t)})$; with $\mathbb{E}\hat{K}_{ii} = 1$ and near-orthogonal values, $\sigma^2 = \sum_{t \neq i} \hat{K}_{ti}^2 / \hat{K}_{ii}^2$ has the stated expectation, and for $\delta \perp v$ with $\|\delta\| = \sigma \|v\|$, $\|(v + \delta) / \|v + \delta\| - v\|^2 = 2 - 2/\sqrt{1 + \sigma^2}$. $\square$

The prediction has no fitted constant; Section 5.10 compares it with the measured ε_1 and tests whether it, together with the single-hop noisy-query curve, predicts two-hop accuracy. Iterating the argument gives the local recursion $\varepsilon_{k+1} \leq \varepsilon_{k+1}^{\text{ct}} + \|W\|_{\text{op}} J \varepsilon_k / \|y\|$: propagated error grows geometrically whenever the output Lipschitz constant exceeds the signal scale, which is the regime we observe for interpolating whitened memories.

4 Experimental setup

Unless stated otherwise $d=64$, $m=256$ ($= 4d$, the upstream default), $R=4$, $\rho=0.1$, five seeds, three test families per seed and configuration (15 independent units), 300 complete queries per hop and family shared across all strategies, and first-hop query noise $\varepsilon \in \{0, 0.1, 0.3, 0.5\}$ added as εu , $u \sim \text{Unif}(S^{d-1})$ (the upstream noisy-query model). Load is varied through $N \in \{64, \dots, 1024\}$ entities per relation (facts per memory $\approx 0.9N$) and through $m \in \{64, \dots, 1024\}$ at $N=512$. Bridges are fitted on three train families; abstention thresholds are the 5% quantile of the confidence statistic over complete tune queries (95% target coverage), frozen before test. Edits use $N=256$, ten rounds of eight random replace/remove/add operations, 400 fixed queries per hop re-labelled after every round, and the affected set is the exact set of queries whose symbolic answer changed. All experiments run on CPU with two threads (float64 build; FP32/BF16 inference in the precision study); total paid compute is 0 USD. Uncertainty: 95% bootstrap intervals over families, paired over families for differences. The protocol (configurations and seeds) was fixed in `scripts/run_full_chain.sh` before the full runs.

Official baseline reproduction. Figure 1 shows the two upstream targets run unmodified on CPU: $\gamma_{\min}$ versus F at $d=64$, $m=512$ (five seeds) and the noisy-query margin at $d=64$, $F=256$, $m=512$ (three seeds). Both reproduce the paper’s qualitative claims (margin decreasing in F with a zero crossing near $F \approx 350$ – 420 ; approximately linear degradation in ε with a zero crossing between $\varepsilon=0.44$ and 0.64). The closed-form noisy bound reported by the upstream code is vacuous at these sizes (large negative), which is one motivation for the data-dependent certificate above.

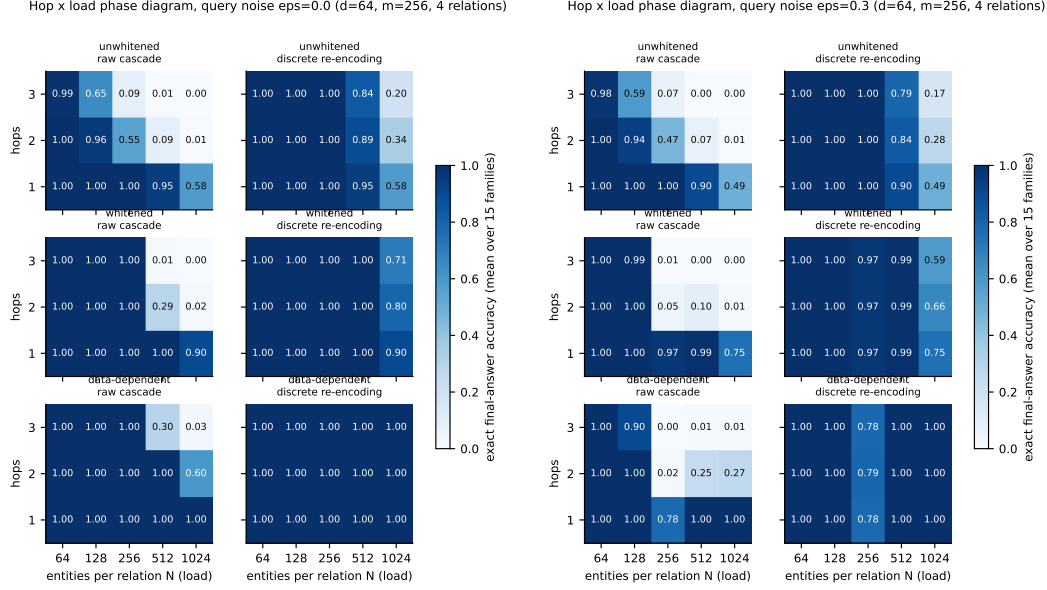


Figure 2: Hop $\times$ load phase diagram at first-hop query noise $\varepsilon=0$ (left) and $\varepsilon=0.3$ (right); rows of panels are the three upstream constructions, columns the raw continuous cascade and discrete re-encoding. Cells show mean exact accuracy over 15 test families ($d=64$, $m=256$, 4 relations).

Table 1: Exact accuracy at $N=512$ entities per relation (≈ 460 facts per memory), $\varepsilon=0$, $d=64$, $m=256$. Mean [95% CI over 15 families]. The symbolic lookup is exact by construction (resource reference only).

construction	strategy	1 hop	2 hops	3 hops
unwhitened	raw cascade	94.6% [94.1, 95.1]	9.0% [7.9, 10.1]	0.5% [0.3, 0.8]
	linear bridge	94.6% [94.1, 95.1]	7.4% [6.6, 8.2]	0.4% [0.2, 0.6]
	MLP bridge	94.6% [94.1, 95.1]	4.9% [4.4, 5.4]	0.2% [0.1, 0.4]
	discrete re-encoding	94.6% [94.1, 95.1]	89.0% [87.9, 90.1]	83.7% [82.6, 84.7]
whitened	raw cascade	100.0% [100.0, 100.0]	28.9% [27.9, 30.0]	0.8% [0.6, 1.1]
	linear bridge	100.0% [100.0, 100.0]	27.9% [26.7, 29.1]	0.8% [0.6, 1.1]
	discrete re-encoding	100.0% [100.0, 100.0]	100.0% [100.0, 100.0]	100.0% [99.9, 100.0]
data-dependent	raw cascade	100.0% [100.0, 100.0]	99.9% [99.8, 100.0]	29.6% [28.4, 30.8]
	linear bridge	100.0% [100.0, 100.0]	99.9% [99.8, 100.0]	28.3% [26.5, 30.0]
	discrete re-encoding	100.0% [100.0, 100.0]	100.0% [100.0, 100.0]	100.0% [100.0, 100.0]
GD bilinear, 10k epochs (equal bytes)	raw cascade	100.0% [100.0, 100.0]	80.6% [79.6, 81.6]	12.2% [11.0, 13.4]
	discrete re-encoding	100.0% [100.0, 100.0]	100.0% [100.0, 100.0]	100.0% [100.0, 100.0]
GD bilinear, 2k epochs (10 fam.)	raw cascade	100.0% [100.0, 100.0]	64.7% [63.7, 65.7]	6.3% [5.6, 7.1]

5 Results

All accuracies are exact final-answer accuracies on held-out composed queries of test families (15 families per cell unless noted), reported as means with 95% family-level bootstrap intervals in brackets. Run 20261021T205349Z_phase_main.

5.1 Hop $\times$ load phase diagram: single-hop success does not compose

Figures 2–3 and Table 1 give the main result. At $N=512$ the unwhitened construction still decodes 94.6% of single-hop queries and the whitened and data-dependent constructions decode 100.0% and 100.0%, i.e. the regime the upstream paper calls “stored”. The raw continuous cascade nevertheless answers only 9.0% (unwhitened) and 28.9% (whitened) of two-hop queries and essentially none of the three-hop queries (0.5%, 0.8%). The data-dependent construction composes exactly at two hops (99.9%) but drops to 29.6% at lower load with every hop: at $N=256$ the unwhitened cascade is at 55.2% / 8.7% for two / three hops while single-hop recall is 100.0%.

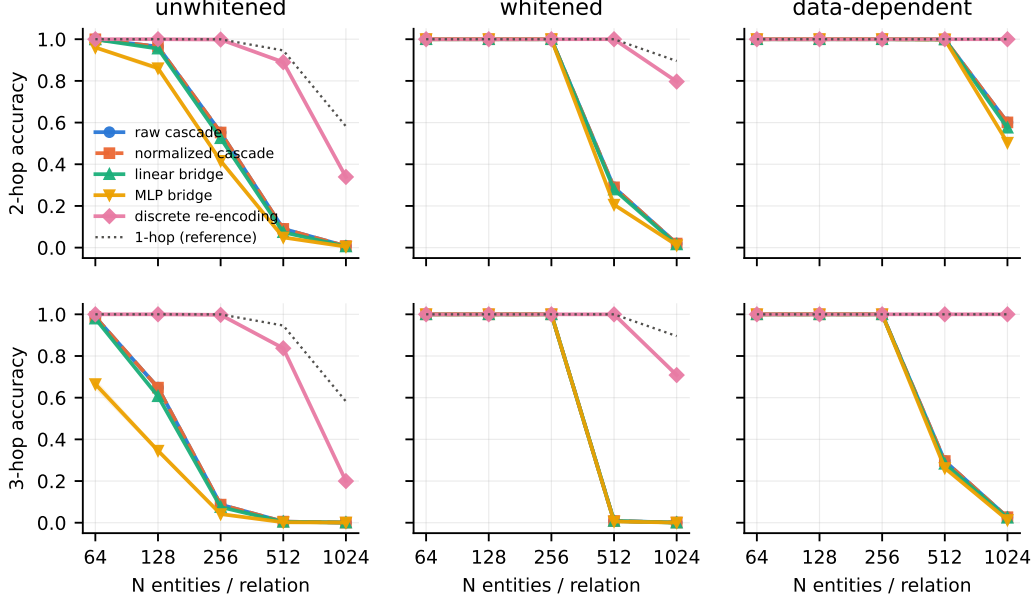


Figure 3: Two- and three-hop accuracy versus load for every composition strategy ($\varepsilon=0$); shaded bands are 95% family bootstrap intervals; the dotted line is single-hop accuracy. Raw, normalized and bridged cascades coincide; only discrete re-encoding tracks the single-hop curve.

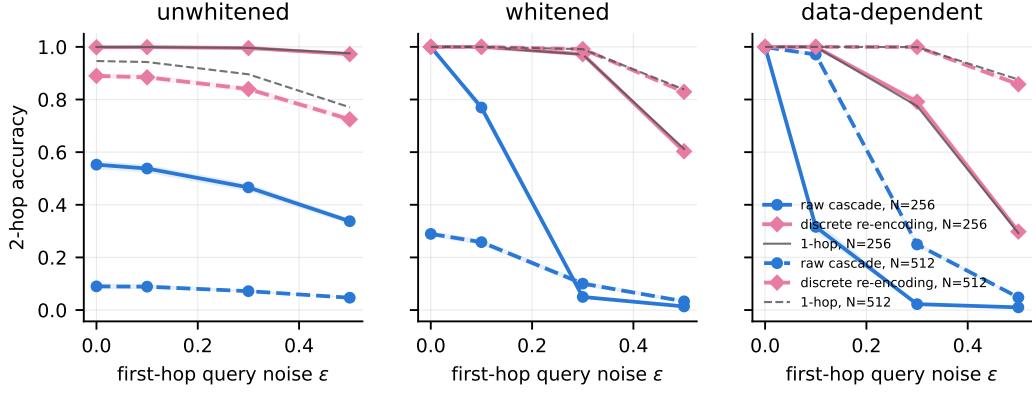


Figure 4: Two-hop accuracy versus first-hop query noise ε at $N=256$ (solid) and $N=512$ (dashed); grey: single-hop accuracy under the same noise.

The equal-byte GD-trained bilinear MLP (upstream recipe, 10,000 epochs), which has much larger unnormalized margins, degrades more gracefully (80.6% at two hops, 12.2% at three) but shows the same pattern (Section 5.5).

The mechanism is propagated cross-talk, not decoding failure at the first hop: the first memory’s output points to the right entity but is far from it (Section 5.7), and the second memory is queried far outside the noise radius its margin tolerates (the upstream noisy-query curve of Figure 1 crosses zero near $\varepsilon \approx 0.5$). Single-hop $\gamma_{\min}$ therefore does not predict multi-hop recall: the whitened memory has $\gamma_{\min} > 0$ at every stored key at $N=512$ and fails at two hops; the unwhitened memory has $\gamma_{\min} < 0$ already at $N=256$ and still composes at 55.2%.

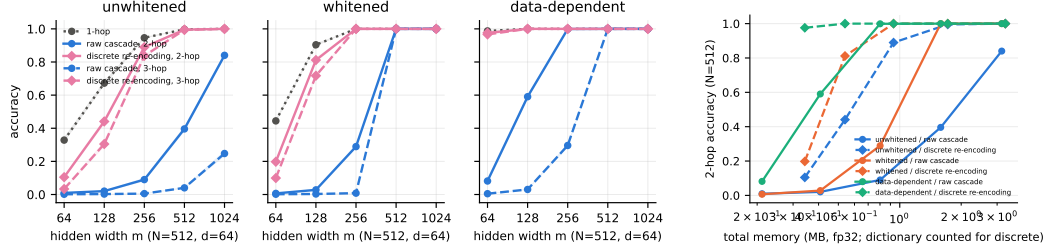


Figure 5: Left: load axis through hidden width m at $N=512$. Right: two-hop accuracy versus total memory bytes (fp32; dictionary counted for discrete re-encoding).

5.2 Query noise

Figure 4 shows the noise axis. The whitened and data-dependent memories at $N=256$ are interpolating ($n < m$, dual branch, ridge 10^{-6}) and amplify query noise: at $\varepsilon=0.3$ their single-hop accuracy is still 97.1% / 77.5% but two-hop accuracy collapses to 5.0% / 2.2%, whereas the same constructions at $N=512$ (over-determined least squares, $n > m$) retain 10.0% / 24.9%. The unwhitened memory is insensitive to this noise level (46.6% vs. 55.2% clean) because its cross-talk already dominates. This is the cross-talk/noise-amplification trade-off predicted by the recursion in Section 3; Section 5.6 shows that it is a property of the interpolation regime, not of the ridge convention.

5.3 Intermediate corrections: normalization, bridges, re-encoding, bytes

Normalization changes no decision (exact-arithmetic identity, Section 3; the precision study below confirms it numerically). Bridges fitted on one-hop pairs from entity-disjoint train families do not help on isotropic entities: paired differences against the raw cascade at $N=512$ are -1.6 [-2.4, -0.9] (linear) and -4.1 [-5.4, -2.8] (MLP) points for the unwhitened memory and -1.1 [-1.6, -0.4] / -8.3 [-9.1, -7.5] for the whitened one; the MLP bridge is slightly *harmful* because it over-fits the train families' entities. This is expected: with isotropic entities, the posterior over the intermediate entity given a noisy direction is centred on that direction, so no map that is independent of the test entity set can reduce the error. Discrete re-encoding against the entity table restores 89.0% / 100.0% / 100.0% two-hop accuracy (paired gains +80.0 [+78.2, +81.7], +71.1 [+70.0, +72.1] points for unwhitened and whitened) and makes multi-hop recall equal to the product of per-hop single-hop recalls, which is the ceiling for any method that only sees primitive edges.

Bytes. At $N=512$ the four relation memories occupy 0.80 MB in fp32; the linear bridge adds a negligible (d^2+d) parameters; the dictionary used by discrete re-encoding is the entity table, which every method already needs for the final decode, so its *incremental* cost is zero and its total cost is 0.93 MB. A direct composed memory for two hops at $N=256$ needs 3.17 MB against 0.79 MB for the primitive memories ($R^2=16$ path memories), grows as R^k , and must be rebuilt after edits (Section 5.8).

Load through width. Figure 5 varies m at fixed $N=512$. Widening the memories moves the raw cascade's two-hop accuracy from 1.0% ($m=64$) to 84.0% ($m=1024$) for the unwhitened construction and from 0.7% to 100.0% for the whitened one, while discrete re-encoding at $m=256$ already gives 89.0% / 100.0%: at matched bytes, spending the dictionary's bytes on width does not buy what re-encoding buys.

Geometry. Two structured entity geometries test whether a bridge can exploit shared structure (Figure 6). With entity tables of all families confined to a shared 16-dimensional subspace of $\mathbb{R}^{64}$ ("low-rank"), the effective load rises sharply (raw two-hop accuracy of the unwhitened cascade at $N=256$ is 1.3% and even discrete re-encoding reaches only 2.8%), and the linear and MLP bridges change nothing (-0.1 [-0.4, +0.2] and -0.4 [-0.8, -0.1] points). This is not a failure to learn the subspace: every memory output is $W\phi(x) = V^T C\phi(x)$, a linear combination of stored values, so both signal and cross-talk already lie in $\text{span}(V)$ and a projector onto it is the identity on the memory's outputs. With clustered entities (16 fresh centres per family, spread 0.3) the unwhitened cascade is

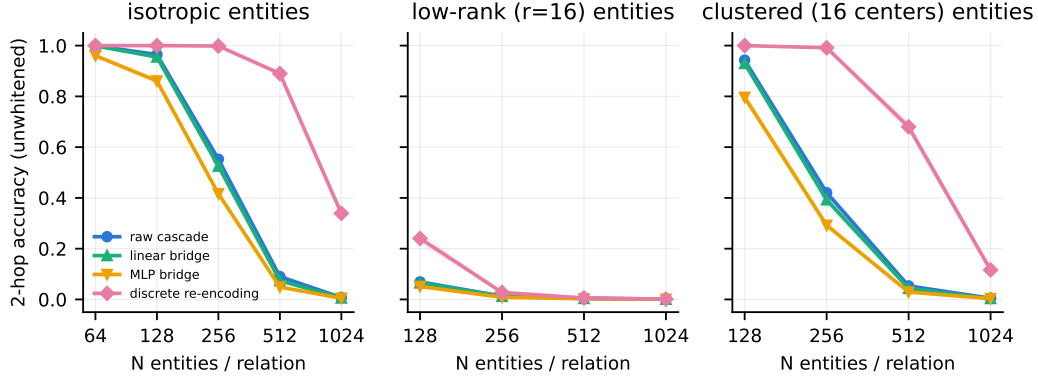


Figure 6: Two-hop accuracy of the unwhitened cascade with bridges under three entity geometries. In no geometry does a bridge trained on entity-disjoint families improve on the raw cascade; low-rank entities mainly raise the effective load.

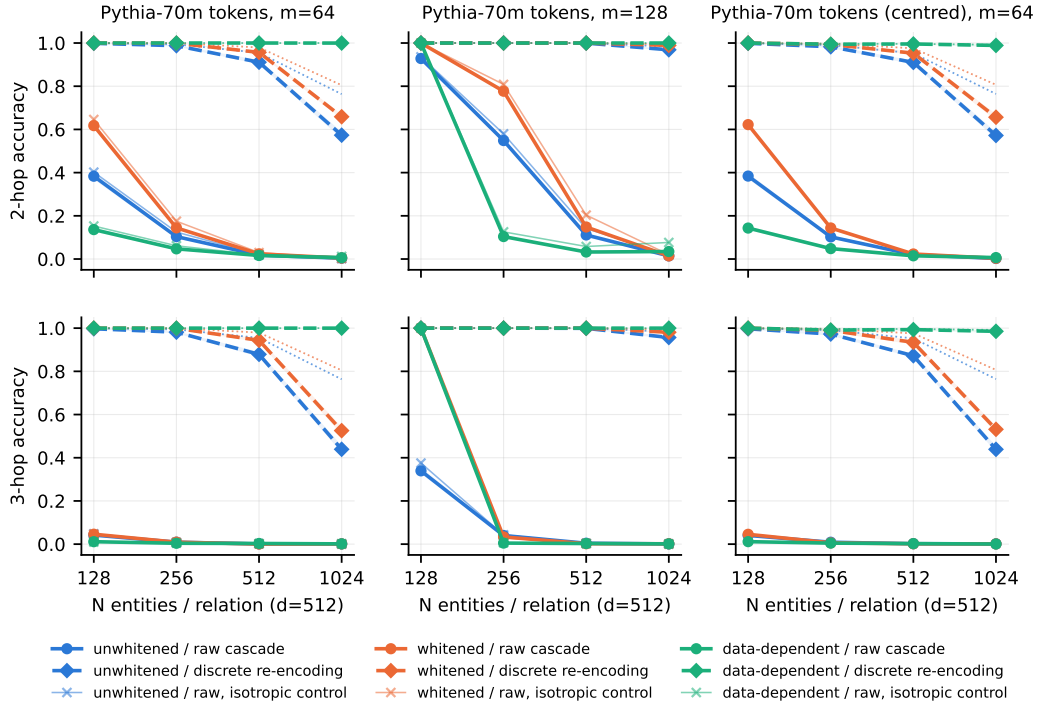


Figure 7: Two- and three-hop accuracy with frozen Pythia-70m token embeddings as entity tables ($d=512$; $\varepsilon=0$; 15 test families per point). Solid: raw cascade; dashed: discrete re-encoding; dotted: single-hop accuracy; thin crosses: isotropic synthetic tables at the same d and m (control). Right: mean-centred embeddings.

at 42.1% at $N=256$ and bridges are again neutral or harmful (-2.8 [-3.6, -2.0] linear, -12.8 [-14.4, -11.3] MLP), while the interpolating whitened and data-dependent memories compose exactly at $N=256$ (100.0%) and fail at $N=512$ (28.5%) as in the isotropic case. In no tested geometry did a bridge trained on entity-disjoint families improve on the raw cascade; the general reason is the span argument above: cross-talk is a mixture of the *other stored values*, which no entity-set-independent map can separate from the target value.

5.4 Frozen real token embeddings

All results above use synthetic unit-norm entity tables. To test whether the conclusions survive a real, anisotropic geometry with confusable entities, we replace the entity tables by rows of the frozen input-embedding matrix of Pythia-70m Biderman et al. [2023] (50,304 tokens $\times$ $d=512$, revision and SHA-256 in `sources.lock.json`), unit-normalised. Each family takes a disjoint slice of one vocabulary permutation, so families remain entity-disjoint. These tables are anisotropic (mean cosine of a token to the vocabulary mean direction 0.065; mean $|\cos|$ of random pairs 0.041 versus $\sqrt{2/(\pi d)} \approx 0.035$ for isotropic vectors) and contain confusable entities: the most similar pair within an $N=1024$ test table has $|\cos| = 0.78$ on average. After mean-centring, a cluster of nearly identical (presumably untrained) tokens appears ($|\cos|$ up to 0.9999). Entity-disjointness is by vocabulary index, so we also measured cross-family near-duplicates, which could only favour the bridges: 0.007% of test entities have a train/tune-family neighbour with $|\cos| > 0.9$ (uncentred) and 0.46% have one above 0.99 (centred). With d fixed by the model, load is varied through N and $m \in \{64, 128\}$. A mean-centred variant tests the mean direction. Because (d, m) now differs from the synthetic main runs ($d=64$, $m=256$), every real run is paired with an *isotropic control* at the same d and m , so that geometry and regime are not confounded.

The phase structure carries over unchanged (Figure 7). At $m=64$, $N=256$ the unwhitened memory decodes 99.4% of single-hop queries but the raw cascade reaches 10.4% at two hops. The whitened cascade reaches 14.4%. The data-dependent cascade reaches 4.7% although it decodes 100.0% of single hops. Discrete re-encoding restores 98.8% / 99.9% / 100.0%. At $m=128$ only the interpolating cell ($N=128$, about 115 facts $< m$) composes for the whitened and data-dependent memories (100.0% / 100.0%); at $N=256$ they fall to 77.7% / 10.4%. Bridges again never help, and on real tables they hurt more: the paired difference to the raw cascade is -33.4 [-34.9, -31.9] points (linear) and -29.7 [-31.2, -28.1] (MLP) for the unwhitened memory at $m=64$, $N=128$, and -59.0 [-60.6, -57.5] (linear) at $m=128$. No bridge cell has a positive mean difference. Mean-centring changes nothing measurable (unwhitened raw two-hop 38.4% centred vs. 38.3% uncentred at $N=128$).

The isotropic control locates the effect of real geometry. At matched (d, m) the raw cascade’s two-hop accuracy on real tables is within a few points of the control (e.g. -2.1 [-3.3, -0.9] points unwhitened and -5.5 [-6.6, -4.4] whitened at $m=128$, $N=512$; real minus isotropic, unpaired family bootstrap). The data-dependent two-hop collapse with perfect single hops appears equally in the control (15.3% at $m=64$, $N=128$). It is therefore a property of the $d \gg m$ regime, not of real embeddings; the synthetic main runs, where this construction composes at $N=512$, sit at $d=64 < m=256$. Real geometry matters mainly through confusable entities at high load. At $m=64$, $N=1024$ it lowers single-hop accuracy by -11.5 [-13.1, -10.0] points (unwhitened) and discrete two-hop accuracy by -19.8 [-21.9, -17.8] points, because re-encoding inherits every first-hop decoding error. In short, on a real embedding geometry single-hop success still does not compose, no dictionary-free bridge helps, and re-encoding against the entity table remains the only correction that works. (We have not analysed the $d \gg m$ mechanism further. A plausible conjecture is that the propagated error has components outside the span of the stored keys, where the data-dependent fit is unconstrained.)

5.5 The gradient-trained competitor at the upstream recipe

Earlier versions trained the equal-byte GD bilinear MLP for 2000 epochs on two test families per seed. Here it uses the upstream GDMLPConfig recipe (Adam $10^{-3} \rightarrow 10^{-6}$ cosine, cross-entropy over the entity table, 10,000 epochs) on all 15 test families. The constructed memories recomputed in the same runs match the main grid exactly (largest difference 0.00 points). Every stored fact is decoded at every load. Longer training improves composition substantially: on the ten families shared with the 2000-epoch runs, raw two-hop accuracy rises by +15.9 [+14.9, +16.8] points at $N=512$ (to 80.6% [79.6, 81.6]) and by +7.7 [+6.5, +8.8] at $N=1024$, and three-hop accuracy by +16.1 [+14.8, +17.4] points at $N=256$. Even so, the learned memory shows the same phase structure. Its raw cascade falls to 18.4% [17.0, 19.8] at two hops and 0.6% [0.3, 0.9] at three for $N=1024$, while discrete re-encoding keeps 100.0% at three hops. Against the constructions, the GD memory beats the unwhitened and whitened raw cascades wherever they carry cross-talk (paired difference +71.6 [+70.6, +72.6] and +51.7 [+50.2, +53.2] points at $N=512$, two hops). It loses to the data-dependent construction (-19.3 [-20.3, -18.3]), whose stored-key outputs are exact. Under first-hop query noise $\varepsilon=0.3$ at $N=256$ it dominates every construction (99.5% [99.3, 99.6] two-hop; +97.2 [+96.8, +97.7] points over data-dependent), because the interpolating constructions amplify noise

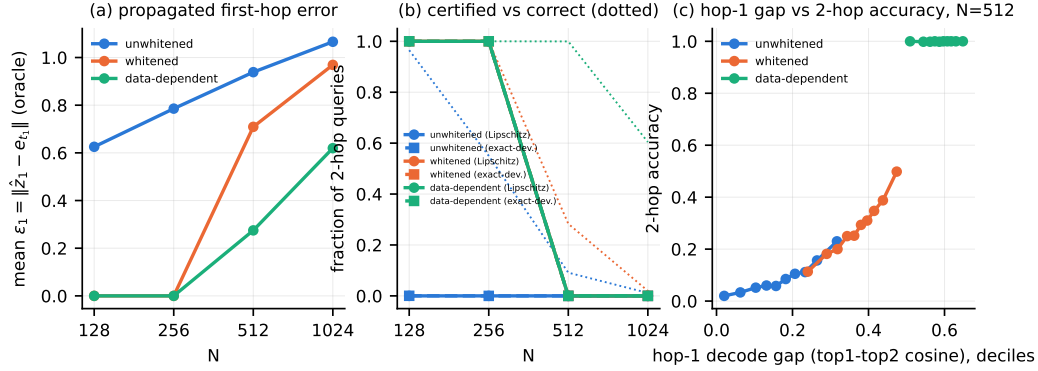


Figure 8: Oracle two-hop diagnostic ($\varepsilon=0$). Left: propagated first-hop error ε_1 . Middle: fraction of queries certified by Proposition 1 (solid) and by the exact-deviation certificate (dashed) against the fraction actually correct (dotted). Right: two-hop accuracy as a function of the first hop’s decode gap (deciles) at $N=512$.

(see the query-noise results above). Gradient training buys margin and noise tolerance at a much higher build cost (108 s versus 10 ms for the four unwhitened memories at $N=512$ on two CPU threads, roughly four orders of magnitude). It does not remove the compositional collapse, and it has no exact local edit rule: an edit means retraining.

5.6 Ridge convention

Replacing the upstream primal/dual readout by a single dual convention (λ for all shapes) changes no cell measurably: the largest paired difference over all constructions, loads, noise levels and hop counts is 0.0 points (Table 3). At the upstream default $\lambda=10^{-6}$ both λ and $n\lambda$ are negligible against the spectrum of $\Phi^\top \Phi$, so the factor- n discontinuity is immaterial here; the regime change at $n=m$ that drives the noise results above is the switch from interpolation ($n < m$) to over-determined least squares ($n > m$), not the ridge convention. We keep the upstream convention for every headline number and note that the convention would matter at larger ridge values, which the paper does not use.

5.7 Does the first hop’s margin predict failure? Certificate and propagated error

Figure 8 quantifies the mechanism with ground-truth intermediates used only for diagnosis. The first-hop output of the unwhitened memory has $\varepsilon_1 \approx 0.79$ at $N=256$ and 0.94 at $N=512$ even when it decodes correctly, far beyond the noise radius the next memory tolerates; interpolating whitened and data-dependent memories have $\varepsilon_1 = 0.00$ at $N=256$ (hence exact composition) but 0.71 at $N=512$. The Lipschitz certificate is sound (0 certified-but-wrong queries among 216,000) and certifies 100.0% of whitened two-hop queries at $N=256$ and 100.0% of data-dependent ones, but 0.0% of unwhitened queries at $N=512$ although 9.1% are correct; the exact-deviation certificate (0 violations) is tighter but still conservative for cross-talk-dominated memories. The first hop’s decode gap is only weakly predictive of final correctness (AUC 0.70 unwhitened, 0.64 whitened at $N=512$): the margin at the first hop is the wrong statistic; the propagated error ε_1 relative to the next memory’s noise tolerance is the right one.

5.8 Edit propagation

Figure 9 shows edit propagation. For the whitened and data-dependent constructions the local rule reproduces full reconstruction: after ten rounds, two-hop accuracy on affected paths is 100.0% (local) versus 100.0% (rebuild) and on unaffected paths 100.0% versus 100.0%, at 1.6 ms versus 6.6 ms per round of eight edits. Because composition is computed at query time, every affected composed answer changes with the primitive edit and nothing else moves; the direct composed memory reaches the same accuracy (100.0% affected, 100.0% unaffected) only by recomputing 57.9 composed facts per round of eight primitive edits for two hops (313.7 for three hops), i.e. $7.2 \times$ the number of ed-

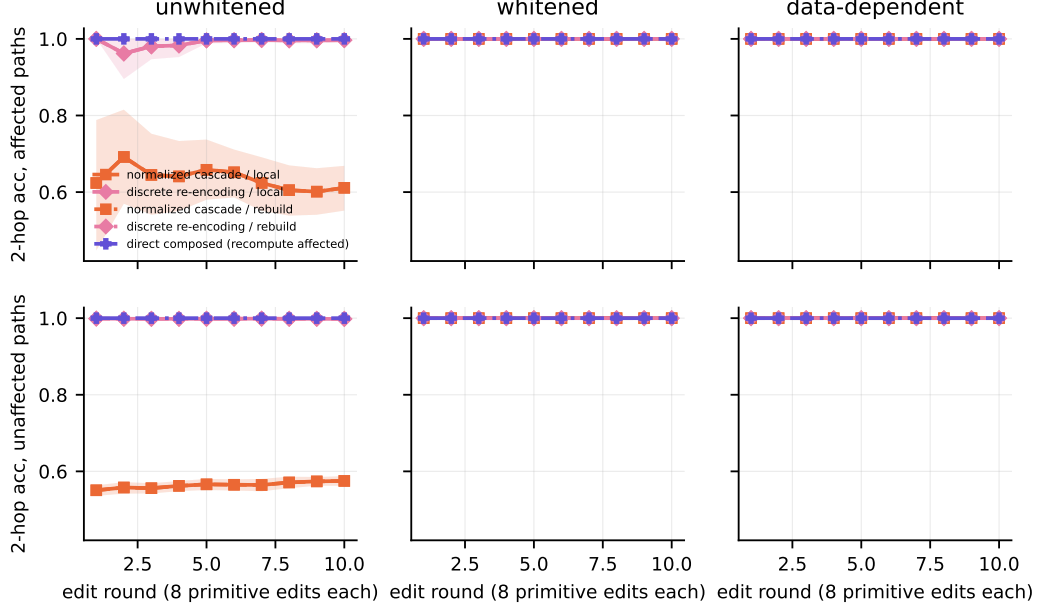


Figure 9: Two-hop accuracy on composed queries whose answer should change (top) and should not change (bottom) over ten rounds of eight primitive edits at $N=256$; local coefficient-row edits (solid) versus full reconstruction (dashed), normalized cascade and discrete re-encoding, and a direct composed memory that recomputes affected composed facts.

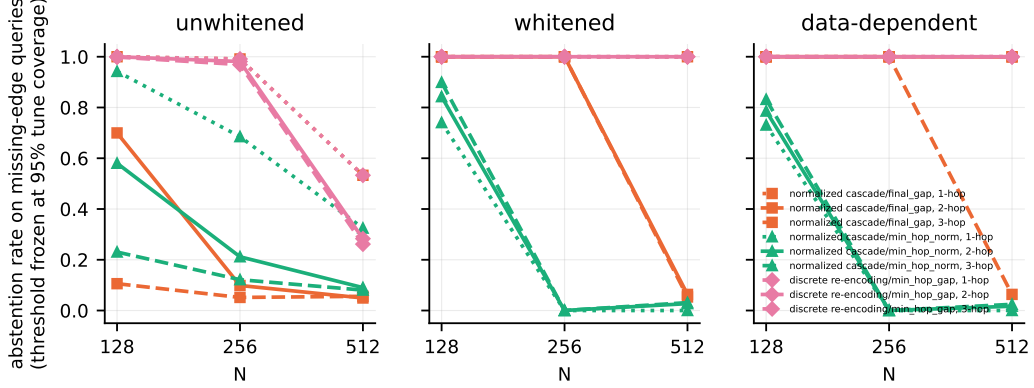


Figure 10: Abstention rate on queries with a missing edge (never scored as recall), with thresholds frozen on tune families at 95% coverage of complete queries, for three confidence statistics and 1–3 hops.

its. For the unwhitened construction, edit propagation is exact in the same sense but limited by its composition accuracy at this load (61.1% affected / 57.5% unaffected with the normalized cascade; 99.7% / 99.8% with discrete re-encoding).

5.9 Abstention on missing edges, and precision

With thresholds frozen on tune families, the final decode gap abstains on 100.0% of missing-edge two-hop queries for the whitened cascade and 100.0% for the data-dependent one at 94.6% / 94.6% coverage of answerable queries; for the unwhitened cascade, whose outputs are cross-talk dominated, it abstains on 9.9% with selective accuracy 61.7%. The table-free statistic $\min_k \|M(z_k / \|z_k\|)\|$ works only for the unwhitened memory (68.5% at one hop) and fails for

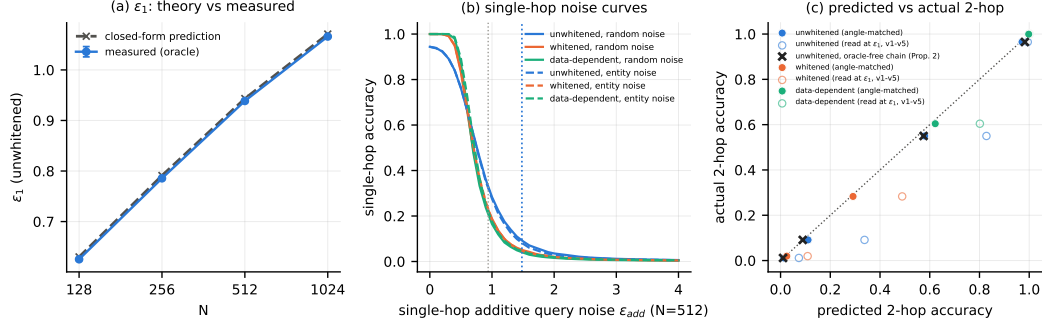


Figure 11: Predictive theory of propagated error. (a) Closed-form ε_1 of Proposition 2 against the measured (oracle) mean ε_1 for the unwhitened memory. (b) Single-hop accuracy under additive random and entity-aligned query noise at $N=512$ (wide grid); blue dotted: the angle-matched $\tan \theta$ at the measured ε_1 ; grey dotted: ε_1 itself (the v2–v5 units error). (c) Two-hop accuracy predicted as $\mathbb{E}_q[\text{acc}_1(\tan \theta_q)]$ (filled), as in v2–v5 at ε_1 (hollow), and by the oracle-free chain $\text{acc}_1(\sigma)$ (crosses), against actual accuracy for every (construction, N).

ridge readouts, whose extrapolated outputs have unit-scale norm; for discrete re-encoding the minimum per-hop decode gap abstains on 98.1% (unwhitened) and 100.0% (whitened) of missing-edge two-hop queries.

Precision. Running inference in float32 or bfloat16 on memories built in float64 changes accuracy by at most 0.24 points in any cell (largest paired difference -0.24 points, CI [-0.60, +0.09], for data-dependent / normalized / bfloat16 / 3-hop / $N=512$). The raw cascade’s intermediate scale shrinks as n^{-2^k} but bfloat16 keeps float32’s exponent range, so neither raw nor normalized cascades lose decisions.

5.10 Predicting multi-hop failure from single-hop quantities

Section 5.7 showed that the first hop’s *margin* is a poor predictor. Proposition 2 supplies a quantity that is predictable: for the unwhitened memory the closed form gives $\varepsilon_1 = 0.630 / 0.791 / 0.943 / 1.071$ at $N = 128/256/512/1024$ against measured 0.625 / 0.786 / 0.939 / 1.066 (Figure 11a), with no fitted constant. The natural next step is to predict two-hop accuracy by reading the next memory’s single-hop noisy-query curve at the propagated error. The two quantities must first be put in the same units. The noisy-query model adds εu to a unit key, whereas ε_1 is a deviation measured after normalisation. Equal angles therefore correspond to $\varepsilon = \tan \theta$ with $\cos \theta = 1 - \varepsilon_1^2/2$, not to $\varepsilon = \varepsilon_1$.

Correction. Versions v2–v5 of this paper read the curve at $\varepsilon = \varepsilon_1$. That units error made the prediction overshoot by 9.2 points on average, and it led us to conclude, wrongly, that propagated cross-talk is more harmful than random noise of equal magnitude. A regression test now pins the mapping.

With the angle-matched mapping and a single-hop noise curve measured on a wider grid ($\varepsilon \leq 4.0$, 28 points), we average the curve over the per-query ε_1 . This predicts two-hop accuracy in all 12 (construction, N) cells with mean absolute error 0.8 points (max 3.0) for random-direction noise and 0.7 points (max 4.1) for entity-aligned noise (Figure 11c). For the unwhitened memory the chain needs no oracle at all. Under its assumptions, Proposition 2’s σ is exactly $\tan \theta$, so we read the single-hop curve at $\varepsilon = \sigma$ ($\sigma = 0.75 / 1.06 / 1.50 / 2.12$). This predicts 98.2% / 57.5% / 8.9% / 1.0% two-hop accuracy at $N = 128/256/512/1024$, against measured 96.5% [95.9, 97.1] / 55.0% [53.9, 56.2] / 9.1% [8.3, 9.9] / 1.1% [0.9, 1.5] (mean absolute error 1.2 points, max 2.5).

Angle-matched interventions on the second hop’s input test the mechanism directly (Table 2). We keep each query’s own angle θ and change only the direction of its deviation: to an isotropic random direction, to a random direction within the span of the stored values, or to the error direction of a different query. None of these changes moves two-hop accuracy by more than 2.7 points in any cell. An oracle-free mean-cross-talk correction (MCC), which subtracts the memory’s average propagated

Table 2: Angle-matched interventions on the second hop’s input (paired difference to the raw cascade in points, mean [95% CI over 15 families]; $d=64$, $m=256$, $\varepsilon=0$). The first three arms keep each query’s own angle θ to the true intermediate and change only the direction of the deviation (oracle diagnostics); MCC subtracts the memory’s mean propagated error (oracle-free, d floats per memory).

construction	N	random direction	in-span random	other query’s error	MCC
unwhitened	256	+2.3 [+1.1, +3.6]	+2.3 [+1.0, +3.6]	+0.5 [-1.2, +2.0]	+2.1 [+1.4, +2.9]
	512	-0.2 [-1.4, +0.9]	+0.1 [-1.0, +1.1]	+1.1 [+0.5, +1.8]	+0.4 [-0.1, +0.9]
	1024	-0.2 [-0.5, +0.1]	-0.1 [-0.5, +0.2]	-0.3 [-0.5, +0.1]	-0.0 [-0.2, +0.2]
whitened	512	-0.8 [-1.9, +0.4]	-0.6 [-2.0, +0.7]	+0.2 [-1.3, +1.7]	-0.1 [-0.3, +0.1]
	1024	-0.0 [-0.5, +0.4]	-0.2 [-0.7, +0.2]	+0.2 [-0.2, +0.6]	+0.0 [-0.1, +0.1]
data-dependent	1024	+1.3 [-0.3, +2.9]	+1.9 [+0.5, +3.3]	+2.7 [+1.0, +4.3]	-0.0 [-0.3, +0.2]

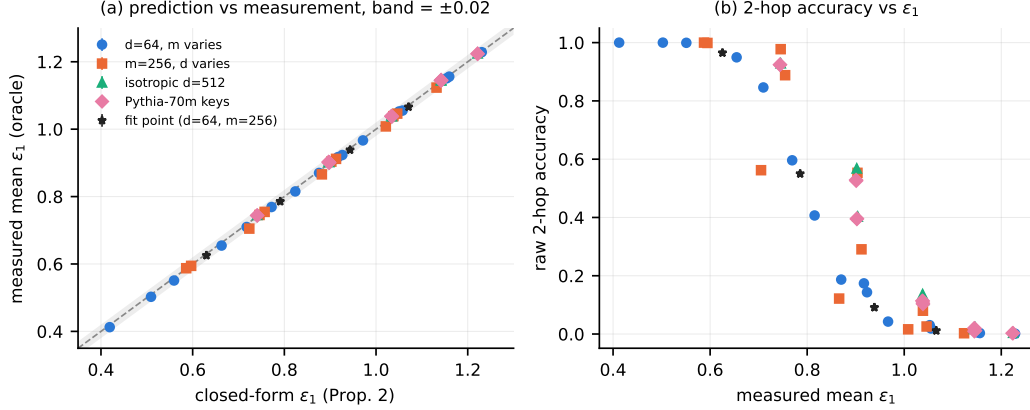


Figure 12: Out-of-sample test of Proposition 2 (unwhitened, $\varepsilon=0$). (a) Closed-form against measured mean ε_1 (95% family intervals) at 13 held-out design points ($d=64$ with $m \in \{64, 128, 512, 1024\}$; $m=256$ with $d \in \{32, 128, 256\}$; isotropic $d=512$, $m \in \{64, 128\}$; Pythia-70m keys, uncentred and centred, $d=512$, $m \in \{64, 128\}$), four loads each; stars mark the $(d, m) = (64, 256)$ point the formula was first checked on; grey band ± 0.02 . (b) Raw two-hop accuracy against measured ε_1 for the same cells.

error (d floats per memory), gains at most +2.1 points. So the second hop treats propagated cross-talk like random query noise at the same angle. What matters is its magnitude, and Proposition 2 predicts that magnitude. The kernel-mean shared bias” mechanism we conjectured in earlier versions is not needed and is not supported. The practical rule is quantitative: a chain of unwhitened memories at load n composes about as well as the next memory answers single-hop queries under additive noise $\sigma(n, d, m)$. This is a prediction from single-hop quantities alone, but it holds only for the constructed memories tested here; the single-hop *margin* remains a poor predictor (Section 5.7).

Out-of-sample test of Proposition 2. The agreement above was observed at the single design point $(d, m) = (64, 256)$. We therefore pre-registered an out-of-sample test (criterion: $|\text{closed form} - \text{measured}| \leq 0.02$ with the family interval inside the ± 0.02 band) at 13 new design points that vary d and m separately and include real keys (Figure 12a). The closed form passes in 34 of 36 isotropic cells (mean absolute error 0.0050). Both failures are at $d=32$; the worst error there is 0.019 ($d=32$, $m=256$, $N=128$), where the approximation $\mathbb{E}\rho^4 \approx 3/d^2$ (exactly $3/(d(d+2))$) is coarsest. A plug-in evaluation of the same second-moment calculation on the actual tables lowers that error to 0.012. Although the real Pythia-70m keys violate the isotropy assumption, the formula passes in 16 of 16 real-key cells (mean absolute error 0.0036; worst 0.006). This explains the matched- (d, m) result of Section 5.4: at $d=512$, $m \leq 128$ the cross-talk is dominated by the geometry-free sketch term $(1+4/d)/m$, not by $\mathbb{E}\rho^4$, so key geometry barely changes ε_1 . Across all cells, ε_1 orders raw two-hop accuracy well (Spearman 0.96 between ε_1 and two-hop error; Figure 12b) but does not determine it: the 13 cells with $\varepsilon_1 \in [0.85, 0.95]$ range from 9.1% to 56.8% two-hop accuracy. The second memory’s noise tolerance, which depends on (d, m, N) , supplies the

rest; this is exactly what the angle-matched chain above combines. Proposition 2 is thus a validated predictor of the first-hop error across d , m , load and one real geometry.

6 Limitations

The main grid is synthetic and constructed: random unit-norm entity embeddings, injective partial relations, and exactly the three upstream constructions plus a GD bilinear MLP of the same shape. The real-geometry test (Section 5.4) uses frozen *input token embeddings* of one small model (Pythia-70m) as entity tables with synthetic relations; it is not a test on contextual activations or on natural-language facts, and it covers only $d=512$ with $m \leq 128$. We did not place multi-layer Transformers around the memories; our “hops” are explicit memory applications, not attention-mediated. The certificate is an oracle diagnostic (it needs the true intermediate) and is loose for raw memories; the exact-deviation variant is tighter but equally oracle. Bridges were trained on one-hop pairs from three families; a larger or differently supervised bridge could behave differently, though the isotropic impossibility argument does not depend on the bridge class. Removal is implemented as zero-value removal, and additions to ridge readouts require a rebuild. Five seeds and fifteen families give interval widths of a few points; small differences within those intervals should not be over-read.

7 Reproducibility statement

Upstream code is pinned (commit 41443978, Apache-2.0) and imported unmodified; `sources.lock.json` records hashes and licenses. Every number in this paper is produced by `scripts/summarize.py` into `paper/numbers.tex` and every figure by `scripts/make_figures.py` from the raw per-query JSONL records in `results/p02/<run_id>/`, each with its configuration, environment, git commit and log. `tests/` contain leakage (entity disjointness; no access to graph or table between hops), aliasing (local edit equals rebuild; removal locality), accounting and certificate-soundness tests. The complete protocol is `scripts/run_full_chain.sh`; it runs on a laptop CPU in a few hours with two threads.

References

- Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning (ICML)*, 2023.
- 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 EMNLP*, 2024.
- Sébastien Bubeck, Ronen Eldan, Yin Tat Lee, and Dan Mikulincer. Network size and size of the weights in memorization with two-layers neural networks. In *Advances in Neural Information Processing Systems*, 2020.
- Vivien Cabannes, Elvis Dohmatob, and Alberto Bietti. Scaling laws for associative memories. In *International Conference on Learning Representations*, 2024.
- Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Xiang Wang, Xiangnan He, and Tat-Seng Chua. AlphaEdit: Null-space constrained knowledge editing for language models. In *International Conference on Learning Representations*, 2025.
- Roberto Garcia, Jerry W. Liu, Ronny Junkins, Atri Rudra, and Christopher Ré. Mlps are hebbians: Constructing efficient fact-storing mlps for transformers. *arXiv preprint arXiv:2607.10034*, 2026. COLM 2026. Code: <https://github.com/HazyResearch/hebbian-mlps>.
- John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

- Teuvo Kohonen. Correlation matrix memories. *IEEE Transactions on Computers*, 100(4):353–359, 1972.
- 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.
- Eshaan Nichani, Jason D. Lee, and Alberto Bietti. Understanding factual recall in transformers via associative memories. *arXiv preprint arXiv:2412.06538*, 2024.
- Hubert Ramsauer, Bernhard Schödl, Johannes Lehner, et al. Hopfield networks is all you need. In *International Conference on Learning Representations*, 2021.
- Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? In *Proceedings of ACL*, 2024.
- Shu Zhong, Mingyu Zeng, et al. Understanding transformer from the perspective of associative memory. *arXiv preprint arXiv:2505.19488*, 2025.
- Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of EMNLP*, 2023.

A Additional tables

B Protocol details

Families: N entities, $R=4$ relations, each a random permutation with a fraction $\rho=0.1$ of edges removed (0.15 in the abstention study). Queries: 300 complete $(h, r_1, \dots, r_k)$ per hop count and family, fixed per family and shared across strategies, variants and noise levels; relations in a path may repeat. Train/tune/test family seeds are $10^5 s + 10^3 \text{role} + i + 1$ for seed s . Bridges: linear ridge regression ($\lambda=10^{-3}$) or a residual GELU MLP (64 hidden, 1500 Adam steps, cosine decay) on 200 (noisy one-hop output, clean tail) pairs per relation from three train families. GD baseline: same bilinear architecture, Adam 10^{-3} with cosine decay to 10^{-6} , 10,000 epochs, cross-entropy over the entity table, all three test families per seed at $N \in \{256, 512, 1024\}$ (main-grid rows: 2000 epochs, two test families per seed). Direct composed memories use the unwhitened construction with $m=256$ per path for $N \leq 256$. Edits: $N=256$, 10 rounds $\times$ 8 operations drawn uniformly from replace/remove/add on random present/absent edges, local policy caches coefficient rows ($n \times m$ floats, counted). Precision: memories built in float64; A_0, A_1, W and the query are cast to the inference dtype. Hardware: Apple Silicon CPU, 2 threads; the full chain takes a few hours; cost 0 USD.

Table 3: Ridge convention ablation: raw cascade accuracy with the upstream primal/dual readout versus a single dual convention.

construction	N	ε	hops	upstream	consistent-dual	diff [95% CI]	fam.
whitened	128	0.0	1	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
whitened	128	0.0	2	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
whitened	128	0.0	3	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
whitened	128	0.3	1	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
whitened	128	0.3	2	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
whitened	128	0.3	3	0.993 [0.990, 0.995]	0.993 [0.990, 0.995]	0.000 [0.000, 0.000]	15
whitened	256	0.0	1	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
whitened	256	0.0	2	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
whitened	256	0.0	3	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
whitened	256	0.3	1	0.971 [0.963, 0.978]	0.971 [0.963, 0.978]	0.000 [0.000, 0.000]	15
whitened	256	0.3	2	0.050 [0.044, 0.056]	0.050 [0.044, 0.056]	0.000 [0.000, 0.000]	15
whitened	256	0.3	3	0.006 [0.004, 0.008]	0.006 [0.004, 0.008]	0.000 [0.000, 0.000]	15
whitened	512	0.0	1	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
whitened	512	0.0	2	0.289 [0.279, 0.300]	0.289 [0.278, 0.300]	-0.000 [-0.001, 0.000]	15
whitened	512	0.0	3	0.008 [0.006, 0.011]	0.008 [0.006, 0.010]	-0.000 [-0.001, 0.000]	15
whitened	512	0.3	1	0.991 [0.989, 0.994]	0.991 [0.989, 0.994]	0.000 [0.000, 0.000]	15
whitened	512	0.3	2	0.100 [0.090, 0.110]	0.100 [0.091, 0.110]	-0.000 [-0.001, 0.001]	15
whitened	512	0.3	3	0.004 [0.003, 0.006]	0.004 [0.003, 0.006]	0.000 [0.000, 0.000]	15
whitened	1024	0.0	1	0.896 [0.887, 0.903]	0.896 [0.887, 0.903]	0.000 [0.000, 0.000]	15
whitened	1024	0.0	2	0.020 [0.016, 0.024]	0.020 [0.016, 0.024]	0.000 [0.000, 0.000]	15
whitened	1024	0.0	3	0.001 [0.000, 0.002]	0.001 [0.000, 0.002]	0.000 [0.000, 0.000]	15
whitened	1024	0.3	1	0.750 [0.736, 0.765]	0.750 [0.736, 0.765]	-0.000 [-0.001, 0.000]	15
whitened	1024	0.3	2	0.011 [0.008, 0.013]	0.011 [0.008, 0.013]	0.000 [0.000, 0.000]	15
whitened	1024	0.3	3	0.001 [0.000, 0.002]	0.001 [0.000, 0.002]	0.000 [0.000, 0.000]	15
data_dependent	128	0.0	1	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
data_dependent	128	0.0	2	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
data_dependent	128	0.0	3	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
data_dependent	128	0.3	1	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
data_dependent	128	0.3	2	0.998 [0.997, 1.000]	0.998 [0.997, 1.000]	0.000 [0.000, 0.000]	15
data_dependent	128	0.3	3	0.898 [0.885, 0.910]	0.898 [0.885, 0.910]	0.000 [0.000, 0.000]	15
data_dependent	256	0.0	1	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
data_dependent	256	0.0	2	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
data_dependent	256	0.0	3	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
data_dependent	256	0.3	1	0.775 [0.760, 0.791]	0.775 [0.760, 0.791]	0.000 [0.000, 0.000]	15
data_dependent	256	0.3	2	0.022 [0.018, 0.027]	0.022 [0.018, 0.027]	0.000 [0.000, 0.000]	15
data_dependent	256	0.3	3	0.003 [0.002, 0.004]	0.003 [0.002, 0.004]	0.000 [0.000, 0.000]	15
data_dependent	512	0.0	1	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
data_dependent	512	0.0	2	0.999 [0.998, 1.000]	0.999 [0.998, 1.000]	0.000 [0.000, 0.000]	15
data_dependent	512	0.0	3	0.296 [0.284, 0.308]	0.296 [0.283, 0.308]	-0.000 [-0.001, 0.000]	15
data_dependent	512	0.3	1	0.999 [0.998, 1.000]	0.999 [0.998, 1.000]	0.000 [0.000, 0.000]	15
data_dependent	512	0.3	2	0.249 [0.240, 0.257]	0.249 [0.240, 0.257]	-0.000 [-0.001, 0.000]	15
data_dependent	512	0.3	3	0.009 [0.007, 0.012]	0.009 [0.007, 0.012]	0.000 [0.000, 0.000]	15
data_dependent	1024	0.0	1	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	15
data_dependent	1024	0.0	2	0.601 [0.591, 0.611]	0.601 [0.590, 0.611]	0.000 [-0.001, 0.001]	15
data_dependent	1024	0.0	3	0.026 [0.020, 0.034]	0.026 [0.020, 0.034]	0.000 [0.000, 0.000]	15
data_dependent	1024	0.3	1	0.999 [0.999, 1.000]	0.999 [0.999, 1.000]	0.000 [0.000, 0.000]	15
data_dependent	1024	0.3	2	0.269 [0.261, 0.278]	0.269 [0.261, 0.278]	0.000 [0.000, 0.000]	15
data_dependent	1024	0.3	3	0.008 [0.006, 0.011]	0.008 [0.006, 0.011]	0.000 [0.000, 0.000]	15

Table 4: Single-hop accuracy and paired gap to raw multi-hop accuracy ($\varepsilon = 0$, $m = 256$).

construction	N	1-hop acc	1-hop - 2-hop raw	1-hop - 3-hop raw	fam.
unwhitened	64	1.000 [1.000, 1.000]	0.000 [0.000, 0.001]	0.012 [0.009, 0.016]	15
unwhitened	128	1.000 [1.000, 1.000]	0.035 [0.030, 0.042]	0.352 [0.334, 0.369]	15
unwhitened	256	1.000 [0.999, 1.000]	0.447 [0.433, 0.464]	0.913 [0.907, 0.919]	15
unwhitened	512	0.946 [0.941, 0.951]	0.856 [0.843, 0.869]	0.941 [0.936, 0.946]	15
unwhitened	1024	0.582 [0.572, 0.592]	0.574 [0.564, 0.584]	0.581 [0.571, 0.592]	15
whitened	64	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	15
whitened	128	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	15
whitened	256	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	15
whitened	512	1.000 [1.000, 1.000]	0.711 [0.700, 0.721]	0.992 [0.989, 0.994]	15
whitened	1024	0.896 [0.887, 0.903]	0.876 [0.867, 0.886]	0.894 [0.886, 0.902]	15
data_dependent	64	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	15
data_dependent	128	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	15
data_dependent	256	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	15
data_dependent	512	1.000 [1.000, 1.000]	0.001 [0.000, 0.002]	0.704 [0.692, 0.716]	15
data_dependent	1024	1.000 [1.000, 1.000]	0.399 [0.389, 0.409]	0.974 [0.966, 0.980]	15