TIDE: Every Layer Knows the Token Beneath the Context

Token Identity Delivered Everywhere

Paper authors: Ajay Jaiswal, Lauren Hannah, Han-Byul Kim,
Duc Hoang, Mehrdad Farajtabar, Minsik Cho (Apple)

Research discussion ▪ arXiv:2605.06216v1
Read on September 14, 2026

A shared token memory supplies identity information at every depth.

Independent reading slides based on the paper PDF and author TeX source.
arxiv.org/abs/2605.06216v1; all page locators use 1-based PDF pages.

TIDE gives the residual stream a direct identity input

Central idea

Learn several static vectors for each token ID, then let every layer decide how much of this memory to add to its contextual computation.

- **Motivation:** rare tokens get fewer lookup updates; some token pairs become hard to distinguish at depth.
- **Evidence:** lower validation loss, larger gains on rare tokens, and higher mean zero-shot scores.
- **Tradeoff:** substantial extra embedding capacity, with relatively small measured decoding overhead.

Paper, Sections 1, 3–4, PDF pp. 1–2, 5–12; Appendix I, p. 23.

The diagnosis concerns update sparsity and representation loss

Rare-token update sparsity

A lookup row is accessed only when its token appears in the input batch.

$$\kappa_v = 1 - (1 - f_v)^{BT}$$

For rare v, $\kappa_v \approx f_v BT$.

Small f_v means few opportunities to learn token-specific parameters.

Contextual collapse

A pair of different tokens can have similar contextual hidden states.

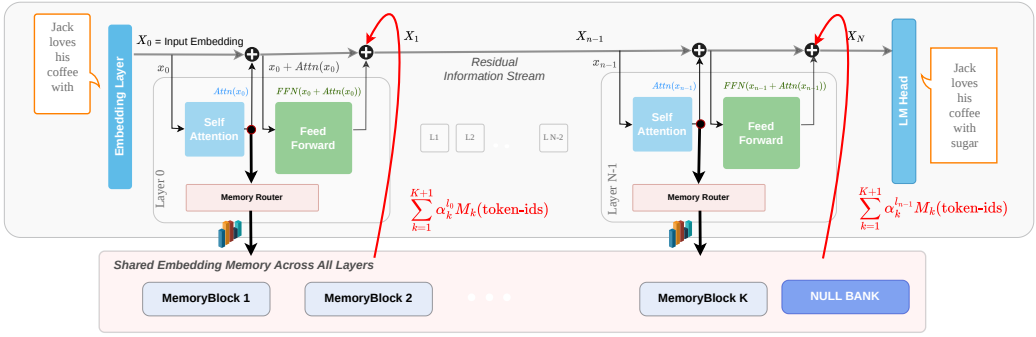
$$\|h_u - h_v\| \leq \delta$$

A fixed-Lipschitz FFN then has limited output separation for that pair.

f_v: token probability; B: batch size; T: sequence length. The probability formula assumes independent positions. Norm diagnostics alone do not prove semantic failure.

Paper, Section 2, PDF pp. 2–4; Appendix C, p. 18. Lookup-only gradient model; tied output embeddings require separate treatment.

One shared memory serves all layers



- Memory lookup uses the **token ID**; routing uses the **current context**.
- Memory and FFN outputs meet through residual addition.

Paper, Fig. 3, PDF p. 5. Original author figure; the formal normalization is specified on the next slides.

Each memory bank learns its own token table

$$E_k \in \mathbb{R}^{|\mathcal{V}| \times d_b}, \qquad M_k(v) = \mathrm{RMSNorm}(E_k[v]), \quad k = 1, \ldots, K$$

$$\mathbf{M} = \mathrm{Stack}_k \, M_k(x) \in \mathbb{R}^{B \times T \times K \times d_b}$$

- The K tables have separate parameters and are shared across depth.
- Compute $\mathbf{M}$ once per forward pass; reuse it at all layers.
- Values are context independent at fixed weights; they are learned during training.
- The presented additive update needs $d_b = d$; no projection for $d_b \neq d$ is specified.

Paper, Eqs. (3.3)–(3.4), PDF pp. 5–6. $|\mathcal{V}|$: vocabulary size; d: model width; d_b: memory width.

The router makes the injected mixture context dependent

$$\tilde{h}^{\ell} = h^{\ell-1} + \mathrm{Attn}(\mathrm{RMSNorm}(h^{\ell-1})), \qquad \tilde{n}^{\ell} = \mathrm{RMSNorm}(\tilde{h}^{\ell}),$$

$$\alpha^{\ell} = \mathrm{softmax}(W_r^{\ell}\tilde{n}^{\ell}), \qquad W_r^{\ell} \in \mathbb{R}^{(K+1)\times d},$$

$$m^{\ell}(v) = \sum_{k=1}^{K+1} \alpha_k^{\ell} M_k(v), \qquad M_{K+1}(v) = 0.$$

The layer update

$$h^{\ell} = \tilde{h}^{\ell} + \mathrm{FFN}(\tilde{n}^{\ell}) + m^{\ell}(v)$$

Null-bank mass suppresses memory. The zero slot has no embedding table, but adds a router output row per layer.

Paper, Eqs. (3.1)–(3.2), (3.5)–(3.6), PDF pp. 5–6. The memory values are static; the routed contribution $m^{\ell}(v)$ is contextual.

The gradient result is a conditional lower bound

Assume that whenever v appears, *every* bank satisfies

$$\left\| \nabla_{e_v^{(k)}} \mathcal{L}_s \right\|^2 \geq G_{\min}^2 > 0.$$

Then summing over banks and steps gives

$$\mathbb{E}\left[\sum_{s=1}^{\tau} \sum_{k=1}^{K} \left\| \nabla_{e_v^{(k)}} \mathcal{L}_s \right\|^2 \right] \geq K \tau \kappa_v G_{\min}^2.$$

- This is K times the **assumed single-bank lower bound**.
- Positive softmax weights do not ensure a uniform nonzero gradient: routing suppression and cancellation remain possible.
- No input occurrence still means no lookup gradient. A K-fold empirical improvement over the baseline does not follow.

Paper, Proposition 3.2, PDF pp. 7–8; Appendix F, p. 21. Appendix simplifies the router to active banks only. Qualifications: reading analysis.

A discrete identity input can bypass a local FFN constraint

For a collapsed pair with target separation $C = \|g(u) - g(v)\|$,

$$\max_{w \in \{u,v\}} \|\mathrm{FFN}(h_w) - g(w)\| \geq \frac{C - L_{\mathrm{FFN}}\delta}{2}.$$

- The bound is informative when $C > L_{\mathrm{FFN}}\delta$, for fixed upstream representations and a controlled Lipschitz constant.
- $E_k[u]$ and $E_k[v]$ are separate rows: their difference need not vanish as $\delta \to 0$.
- This supplies **capacity to separate**; it does not prove optimization success or useful separation after routing.

Normalization caveat

For fixed RMSNorm gain, memory norms are bounded. The paper's "arbitrary C" construction needs additional scale freedom.

Paper, Propositions 2.2 and 3.3, PDF pp. 4, 8; Appendices D/G, pp. 19–20, 22. Caveats: reading analysis.

Training compares a fixed backbone with added memory

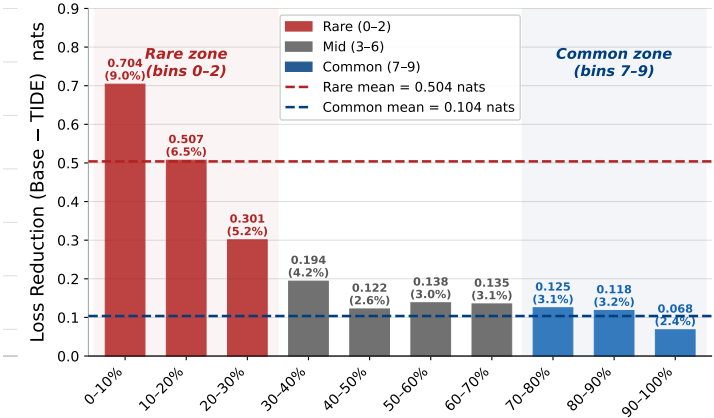
- **Model labels:** 750M, 1B, and 3B LLaMA-style backbone families in Table 2; TIDE adds K embedding tables.
- **Training:** DCLM; LLaMA-3.1 tokenizer, 128,256 entries; sequence length 2,048; Adam and cosine learning-rate schedule.
- **Budgets:** Fig. 5 compares 200B tokens; Fig. 8 extends to 500B. Table 7 lists 400–500B for the general configuration.
- **Evaluation:** WikiText-2, PubMed, DCLM perplexity; eight zero-shot benchmarks; frequency-bin and layer diagnostics.

Comparison boundary

The added memory changes total parameter capacity. These are not demonstrated equal-total-parameter or equal-training-compute comparisons.

Paper, Table 2 / Fig. 8, PDF p. 10; Appendix L / Table 7, pp. 26–27. Checkpoint-to-budget mapping is incompletely documented.

Loss gains are concentrated on rare token types



TIDE-8E-1B vs. Base
Matched 200B tokens.

Mean loss reduction:

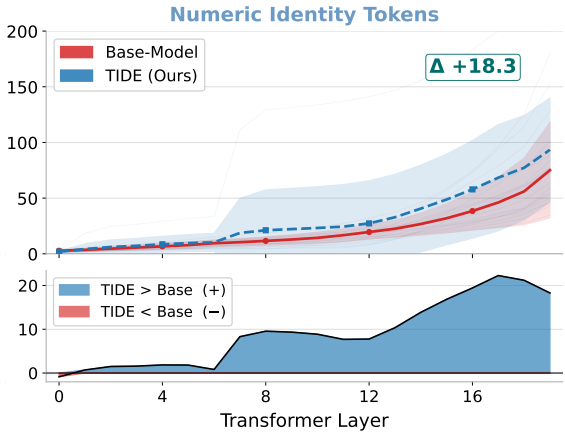
- Rare: 0.504 nats.
- Common: 0.104 nats.

Rare/common ratio:
$\approx 4.8\times$.

All deciles improve; reductions are not strictly monotone.

Paper, Fig. 5 right panel (cropped from original), PDF p. 8. Rare = bins 0–2; common = 7–9; bins contain similar numbers of token types, not occurrences.

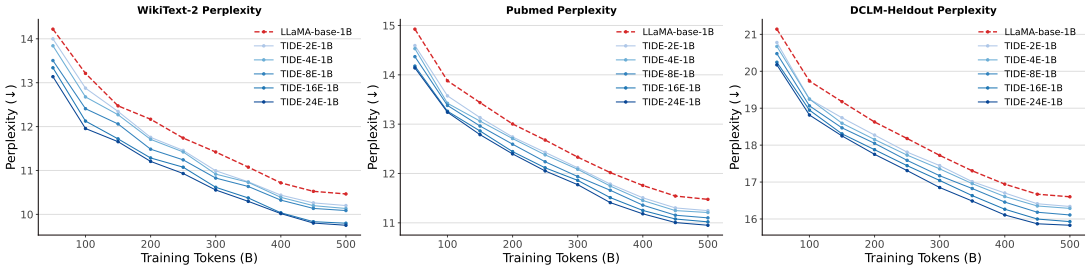
Numeric token pairs gain separation at middle and late layers



- Numerical pairs gain separation from middle layers onward.
- Other panels gain separation late; rare-domain pairs lose it at some early layers.
- Raw ℓ_2 is diagnostic; larger norms can increase it.

Paper, Fig. 6 numeric-token panels, PDF p. 9, cropped original. Top: raw pairwise ℓ_2; bottom: TIDE − Base. Template count differs: 150 in text vs. 250 in Fig. 2 caption.

Validation perplexity generally falls as memory capacity grows



- Reported TIDE curves outperform the baseline on all three corpora.
- Larger K generally helps; small local departures from ordering exist.
- The plot does not establish the claimed $2\times$ token efficiency for $K = 2$–4.

Paper, Fig. 8, PDF p. 10. Original figure; qualification based on plotted values. More training-efficient per token does not imply less total training compute.

The nominal 1B family reports a 2.3 pp average gain

Model	PPL ↓	BoolQ	HellaSwag	PIQA	Reported avg. ↑
Base	5.19	61.7	63.9	74.9	61.4
TIDE, $K = 2$	4.97	68.7	64.9	75.5	62.6
TIDE, $K = 8$	4.89	69.3	65.3	75.5	63.0
TIDE, $K = 16$	4.78	69.7	65.3	75.9	63.3
TIDE, $K = 24$	**4.60**	69.5	66.3	77.3	**63.7**

- Reported eight-task average: 61.4 → 63.7 (+2.3 percentage points).
- Re-averaging the published task entries gives +2.2 percentage points.
- PPL improves by 11.4%; individual tasks need not improve with each K.

Paper, Table 2, PDF p. 10; selected columns, transcribed. Scores in %; BoolQ uses accuracy, HellaSwag/PIQA normalized accuracy. Source caption calls PPL "LAMBDA"; dataset naming is ambiguous.

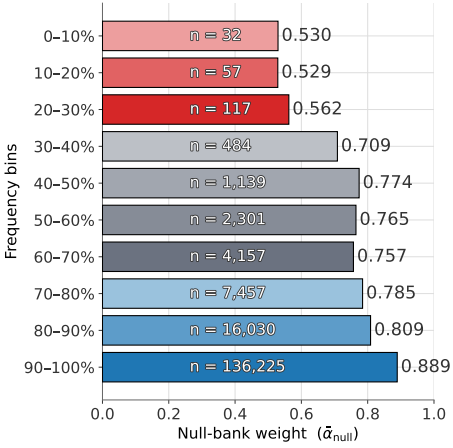
Average gains span three families, with task-level regressions

Backbone label	Base avg.	TIDE-8E avg.	Gain (pp)	PPL: Base → TIDE
750M	59.7	60.7	+1.0	5.63 → 5.18
1B	61.4	63.0	+1.6	5.19 → 4.89
3B	67.2	68.3	+1.1	4.00 → 3.86

- Examples of regressions: 750M BoolQ 63.5 → 63.0; 3B SciQ 93.3 → 93.2.
- Within the nominal 1B family, only HellaSwag and PIQA are nondecreasing across *all* reported K, including Base.
- No multi-seed variance or confidence intervals are reported for Table 2.

Paper, Table 2, PDF p. 10. Exact score transcriptions; gains calculated from rounded source averages. Backbone labels exclude the added table capacity.

The last-layer router uses more active memory for rare tokens

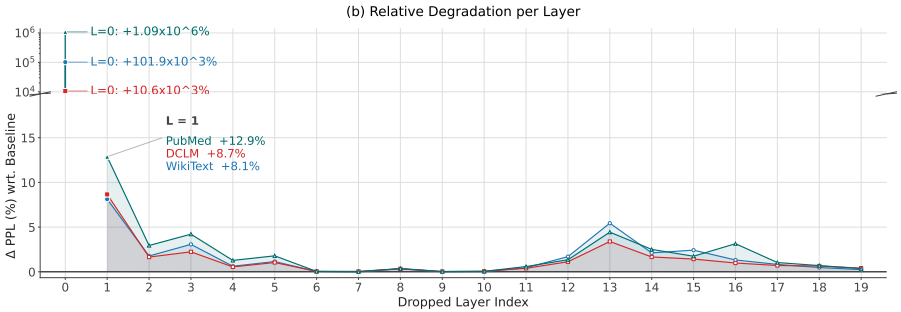


- Null mass rises *overall* from 0.530 in the rarest bin to 0.889 in the most frequent bin.
- Active mass is therefore 0.470 vs. 0.111.
- Small reversals appear between bins; the trend is not strictly monotone.
- This is a last-layer observation; it does not establish routing behavior at every depth.

Complementary bank representations are also suggested by cosine distances and nearest-neighbor examples.

Paper, Fig. 10 right panel, PDF p. 12 (original); Fig. 9, p. 11; Appendix K, pp. 25–26.

Memory dependence varies across depth



Blue: WikiText-2 Red: DCLM Teal: PubMed

- Zero one layer's memory at inference; retain the other injections.
- Layer 0 removal is catastrophic; layer 1 costs 8.1–12.9% PPL.
- Middle layers have smaller marginal effects; layer 13 has a secondary peak.

Paper, Fig. 11(b), PDF p. 22, cropped original; Appendix H, p. 23. This tests dependence of a trained checkpoint, not the optimal injection schedule after retraining.

Embedding memory adds billions of stored parameters

$$P_{\text{tables}} = K|\mathcal{V}|d_b, \qquad P_{\text{routers}} = L(K+1)d.$$

K	Added table parameters	FP16 storage	4-bit storage
2	0.525B	1.051 GB	0.263 GB
8	2.101B	4.203 GB	1.051 GB
24	6.304B	12.608 GB	3.152 GB

- Calculated for $|\mathcal{V}| = 128{,}256$, $d_b = d = 2{,}048$; table storage only, decimal GB, excluding metadata.
- A "TIDE-8E-1B" label therefore does **not** mean 1B total parameters.
- Static lookup permits caching/offloading in principle; runtime activations and training optimizer state also require memory.

Paper, Section 3.2 / Fig. 4, PDF p. 6; Appendix J, pp. 23–24. Counts calculated from the reported vocabulary and width.

Measured decoding overhead ranges from 1.4% to 21.1%

Model (nominal 1B family)	ms/token ↓	Increase vs. Base
Base	11.085	—
TIDE, $K = 2$	11.236	1.4%
TIDE, $K = 4$	11.854	6.9%
TIDE, $K = 8$	12.688	14.5%
TIDE, $K = 16$	12.901	16.4%
TIDE, $K = 24$	13.422	21.1%

- Reported on one B200 GPU, averaged over 5,120 generated tokens.
- Table 4 does not document an SSD-offload benchmark; these timings cannot validate its end-to-end cost.

Paper, Table 4 / Appendix I, PDF p. 23. Times transcribed; relative overhead calculated as (TIDE/Base $-1)\times 100\%$.

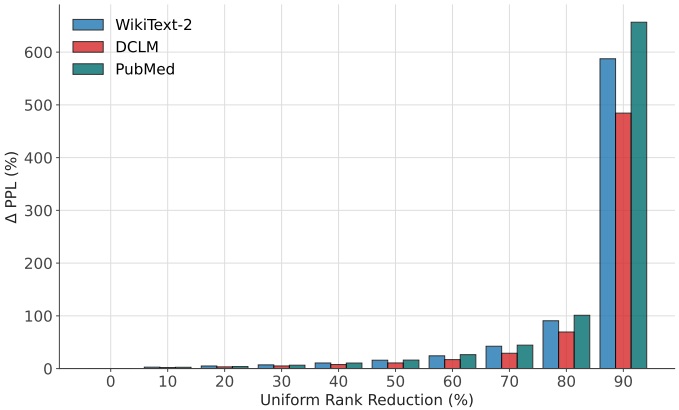
8-bit table quantization changes perplexity very little

Table precision	Size	WikiText-2	PubMed	DCLM
16-bit	4.20 GB	10.088	11.100	16.108
8-bit	2.10 GB	10.089	11.100	16.113
4-bit	1.05 GB	10.263	11.277	16.343

- Results are for the memory tables of TIDE-8E-1B; lower PPL is better.
- At 4 bits, PPL rises by 1.73%, 1.59%, and 1.46%, respectively.
- This supports a useful storage tradeoff at this checkpoint, with a small measurable quality cost.

Paper, Table 5 / Appendix J.1, PDF p. 24. Exact table values; percentage changes calculated relative to 16-bit.

Low-rank compression trades quality for storage



Blue: WikiText-2 Red: DCLM
Teal: PubMed

- Uniform SVD on eight tables: $r = \lceil (1-p)d_b \rceil$.
- 10–30% rank reduction has small PPL impact.
- At 50%, PPL rises roughly 10–20%.
- At 90%, PPL rises by 484–657%.

Paper, Fig. 12(b) / Appendix J.2, PDF pp. 24–25. Cropped original; the 50% range is a visual estimate. p is the rank reduction fraction.

Next tests should separate architecture from extra capacity

1. **Capacity and compute controls:** compare equal-total-parameter and equal-training-FLOP alternatives, including a single enlarged table.
2. **Causal diagnostics:** measure normalized separation, gradient magnitudes, and the effect of token-ID intervention.
3. **System validation:** report SSD latency, cache hit rates, activation memory, and throughput under realistic batches.
4. **Reproducibility:** resolve layer-count and training-budget inconsistencies, and report multiple seeds and checkpoint details.

These are proposed follow-up experiments. They are not results reported in the paper.

Reading analysis based on Sections 3–4 and Appendices H–M. Author limitations include storage, longer training, instruction tuning/RLHF, and bank interpretability (PDF p. 26).

The useful contribution is a persistent token-indexed side input

Take-home message

A learned static memory, selected by contextual routers, is a promising way to preserve token-specific capacity across transformer depth.

- **Supported:** rare-token loss benefits, improved mean benchmarks, and a quantizable memory component.
- **Qualified:** the theoretical gradient claim is conditional; storage increases substantially; causal and systems claims need stronger controls.
- **Discussion:** how much of the benefit comes from repeated identity access, and how much from added token-specific capacity?

Ajay Jaiswal et al. *TIDE: Every Layer Knows the Token Beneath the Context*. arXiv:2605.06216v1, 2026.
Versioned paper; original PDF and author TeX source read. Main talk ends here; five backup slides follow.

Backup: the null-bank proof needs a realizable constant logit

With memory norms bounded by C_M, the algebra is valid:

$$\left\| m^{\ell}(v) \right\| \leq (1 - \alpha_{K+1}^{\ell}) C_M.$$

If active logits are zero and the null logit equals s,

$$1 - \alpha_{K+1}^{\ell} = \frac{K}{K + e^{s}} \longrightarrow 0.$$

- The stated router is bias-free: $z = W_r^{\ell} \tilde{n}^{\ell}$.
- One fixed linear map cannot generally impose the same positive null logit for all possible hidden states; e.g., n and $-n$.
- A bias/constant feature or a suitable restricted hidden-state domain would justify the construction. The paper does not establish this condition.

Paper, Proposition 3.1, PDF p. 7; Appendix E, pp. 20–21. Gap identified in reading analysis; it concerns the stated router-based proof, not impossibility of suppressing memory by other parameter settings.

Backup: occurrence counts are not update-step counts

For 200B training tokens, $B = 8$, and $T = 2{,}048$,

$$\tau = \frac{200 \times 10^9}{8 \times 2{,}048} \approx 1.22 \times 10^7.$$

The expected number of steps containing token v is

$$\mathbb{E}[N_v] = \tau[1 - (1 - f_v)^{BT}] \leq \tau.$$

- Table 1 lists 1.66×10^9 "non-zero gradient updates" for a common token with $f_v = 8.3 \times 10^{-3}$.
- That value is the expected *occurrence count* $200\text{B}\, f_v$; it exceeds the number of optimizer steps.
- The approximation $\tau f_v BT$ is valid only when $f_v BT \ll 1$. It works for rare tokens, and fails for that common token.

Paper, Table 1 / Eq. (2.2), PDF p. 3; Appendix C, p. 18. Arithmetic and interpretation checked in this reading.

Backup: a softmax does not guarantee K times gradient energy

A simplified counterexample to an unconditional claim

Let $m = \frac{1}{K}\sum_{k=1}^{K} e_k$ and let the upstream gradient be $g = \nabla_m \mathcal{L}$. Then

$$\nabla_{e_k}\mathcal{L} = \frac{g}{K}, \qquad \sum_{k=1}^{K} \|\nabla_{e_k}\mathcal{L}\|^2 = \frac{\|g\|^2}{K}.$$

- Every route is positive, but total squared gradient decreases as K grows in this example.
- Therefore a K-independent lower bound $G_{\min}$ is a substantive assumption, not a consequence of positive routing alone.
- This toy example isolates routing; it is not a simulation of trained TIDE.

Teaching derivation by the reader; relates to Proposition 3.2 and Appendix F, PDF pp. 7–8, 21. It illustrates the limitation of an unconditional interpretation.

Backup: two banks capture half of the rare-token gain

Selected cross-entropy values (nats; lower is better):

K	0	2	4	8	16	24
Rare	6.671	6.454	6.361	6.353	6.254	6.250
Common	2.967	2.940	2.928	2.929	2.893	2.892

- $K = 0 \to 24$: rare loss falls by 0.421 nats; common loss by 0.075.
- $K = 2$ captures 51.5% of the total rare gain; $K = 16 \to 24$ adds only 0.004 nats.
- The common curve has a small $K = 4 \to 8$ reversal.

Different evaluation views. Fig. 5: fixed $K = 8$, 200B tokens, individual deciles. Fig. 7: varying K, grouped bins. Keep their values separate.

Paper, Fig. 7, PDF p. 9; Section 4.1, pp. 10–11. Values transcribed from the figure; differences and ratios calculated in this reading.

Backup: banks differ from the input table in neighbor structure

Embedding-space distance

For TIDE-8E-1B, mean row-wise cosine distance between the primary table E and each M_k ranges from 0.649 to 0.990.

- The measured coordinates differ from E; distance alone does not establish useful new semantics.

Nearest-neighbor probe

Top-10 cosine-neighbor sets have lower Jaccard overlap with E for rare tokens than common tokens. For `asynchronously`, examples include `callback` and `caching` in memory banks.

- Different neighbors suggest complementary information, but there is no blinded semantic-quality assessment in this probe.

Paper, Fig. 9, PDF p. 11; Fig. 13, p. 25; Table 6, p. 26 (Section 4.2 / Appendix K). Interpretation: reading analysis.