Copy-Paste Language Models

A direct output and gradient path for in-context copying

Paper by Nathan Godey and Yoav Artzi

Research group discussion

arXiv:2610.19247v1 23 October 2026

A lightweight pointer head improves pretraining and long-context access across tested backbones.

Copying is common, but the output layer makes it expensive to learn

- About **60%** of FineWeb-Edu tokens have appeared earlier in their 4k-token context.
- Self-attention can retrieve a past token; the final prediction still passes through a fixed vocabulary projection $W \in \mathbb{R}^{V \times D}$.
- With $V \gg D$, this head limits output expressivity and compresses the backward signal before it reaches the backbone.

Research question

Can a direct pointer to previous positions improve modern decoder-only LM pretraining, even when the backbone already has attention?

Paper, Secs. 1–2.1, PDF pp. 1–3.

The contribution is a second prediction *and* gradient path

Standard LM head

Predicts over the fixed vocabulary $\mathcal{V}$ using $w_v^\top h_t$. Its backward projection maps vocabulary-logit gradients to D hidden dimensions.

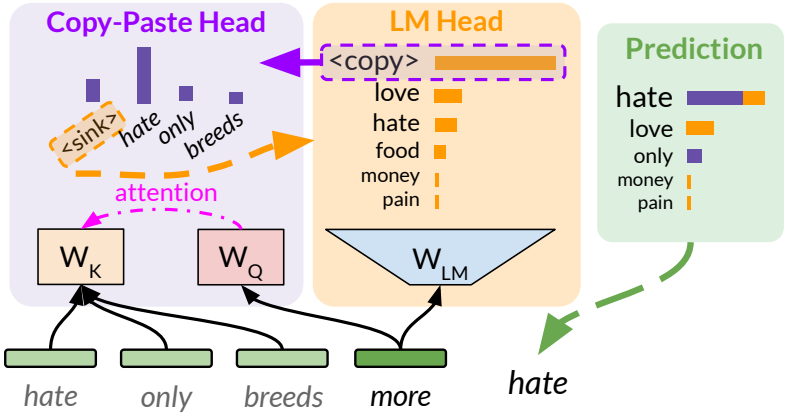
Copy-Paste head

Predicts over context positions $1{:}t$. Its target set changes with the context and provides a shorter gradient projection when $t \ll V$.

Point to a position $\rightarrow$ recover its token $\rightarrow$ mix with the LM distribution.

Paper, Secs. 1–2.2, PDF pp. 2–3. Pointer mechanisms themselves predate this work.

One extra head points at previous positions



- <copy> gates the pointer; <sink> lets the pointer defer to the LM head.

Paper, Fig. 1 (left), PDF p. 1. Original author figure.

The pointer distribution becomes a token distribution

$$q_t = W_Q h_t, \qquad k_\tau = W_K h_\tau,$$

$$\alpha_t(\tau) = \mathrm{softmax}_{\tau' \le t}\left(\frac{k_{\tau'}^\top q_t}{\sqrt{D}}\right)_\tau,$$

$$p_{\mathrm{CP}}(v \mid x_{\le t}) = \sum_{\tau \le t:\, x_\tau = v} \alpha_t(\tau).$$

- A token appearing twice receives the sum of both position probabilities.
- An unseen token receives zero CP probability and remains available through p_{LM}.

Paper, Sec. 2.2, Eqs. (2)–(3), PDF p. 3. Default query/key dimension is D.

The gate and sink form a valid mixture over the vocabulary

$$p(v \mid x_{\leq t}) = (1 - \lambda_t) p_{\mathrm{LM}}(v \mid x_{\leq t}) + \lambda_t p_{\mathrm{CP}}(v \mid x_{\leq t}), \qquad \lambda_t = c_t(1 - s_t)$$

- c_t is the LM head probability of a special <copy> token.
- s_t is CP attention on a learned <sink> key.
- p_{LM} is normalized over the *real* vocabulary, without <copy>.

Training signal

Only standard next-token cross-entropy on p; no auxiliary loss or labeled copy decisions.

Paper, Sec. 2.2, Eqs. (4)–(7), PDF p. 3.

The gate also makes exact conditional sampling possible

1. Sample from the LM head over $\mathcal{V} \cup \{\texttt{<copy>}\}$.
2. If <copy> is sampled, compute copy attention over cached keys.
3. If the sink is sampled, fall back to the normalized LM distribution.

Implementation consequence

Copy attention can be skipped on sampling steps that do not choose <copy>. At inference, one additional copy key is cached per token.

- Greedy decoding still requires the argmax of the *mixture*; independent argmax decisions are not equivalent.

Paper, Appendix B and C.4, PDF pp. 19, 23.

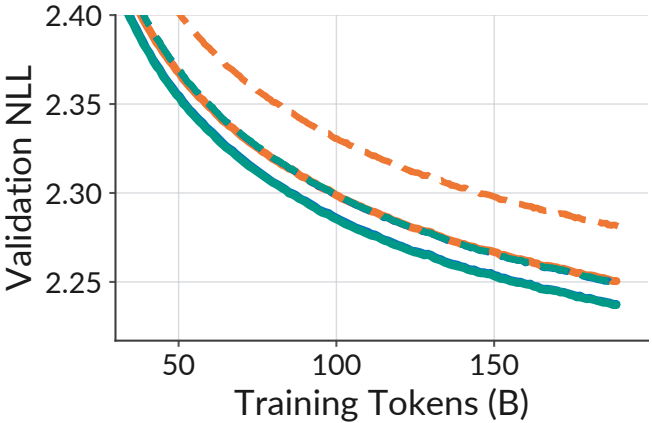
Matched comparisons span scale and backbone context access

Scale	Training tokens	Train length	Hidden size
135M	20B	2,048	576
360M	20B	2,048	960
1.7B	200B	4,096	2,048

- FineWeb-Edu pretraining; each vanilla/CPLM pair uses the same recipe.
- Full attention, sliding-window (SW), and 3:1 Hybrid backbones; SW/Hybrid use a 256-token window in the main 1.7B comparison.
- Hybrid provides global backbone access as a comparison to CP-mediated global access.

Paper, Sec. 3.1 and Appendix A, PDF pp. 4, 18.

CPLMs reach the vanilla final loss with fewer tokens



Paper, Fig. 2, PDF p. 4. Curves show validation NLL before LR decay.

- 1.7B, 200B-token runs.
- **Full / Hybrid:** about 1.25× fewer tokens to the vanilla final loss.
- **SW:** about 2× fewer tokens.
- Solid: CPLM; dashed: vanilla. Blue: Full; green: Hybrid; orange: SW.

Base-model accuracy improves modestly; LAMBADA gains most

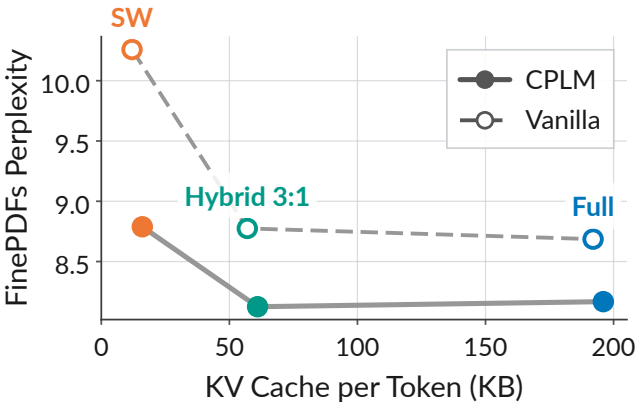
1.7B zero-shot average (%)	Vanilla	CPLM	Gain (pt)
Full	62.45	**63.06**	+0.61
SW-256	61.63	**62.48**	+0.85
Hybrid 3:1	62.22	**62.59**	+0.37

Task with a clear copy opportunity

LAMBADA: Full 44.46 $\rightarrow$ 46.92; SW 42.46 $\rightarrow$ 46.42. The target word often appeared earlier in the passage.

Paper, Sec. 3.1, Table 1, PDF p. 4. Average of seven benchmark columns.

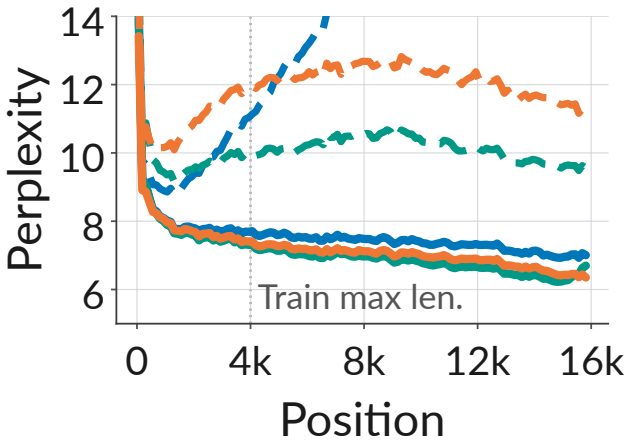
The copy cache shifts the perplexity–memory frontier



Paper, Fig. 1 (right), PDF p. 1. Original author plot.

- On FinePDFs, 1.7B CPLMs lower PPL by about 0.5–1.5 points against paired vanilla backbones.
- SW CPLM matches the Full vanilla model at about **12× less cache per token**, including its copy cache.
- Lower and further left is better.

CPLMs keep exploiting document context beyond train length



- 5,000 FinePDFs documents, 16k tokens each; train length was 4k.
- CPLM PPL falls to about 6.5–7 at 16k; vanilla SW/Hybrid plateau around 10–13.
- SW CPLM copies from over 2,000 tokens back on average at position 16k, beyond its 256-token window.

Paper, Fig. 4, PDF p. 5; Appendix G.2, Fig. 15, PDF p. 28. Full/Hybrid use YaRN $\times 4$.

Supervised fine-tuning preserves the gains

1.7B SFT task average	Vanilla	CPLM	Gain (pt)
Full	24.2	**26.4**	+2.2
Hybrid 3:1	24.3	**25.5**	+1.2
SW-256	18.9	**24.0**	+5.1

- Tulu-3 SFT, up to 16k context; math, code and instruction evaluations.
- SW: MATH 7.2 $\rightarrow$ 16.4, GSM8K 18.6 $\rightarrow$ 25.2.
- Hybrid MBPP is an exception: 19.0 $\rightarrow$ 17.4.

Paper, Sec. 3.2, Table 2, PDF p. 6. MATH/GSM8K use pass@4; code uses pass@10; see Appendix D.

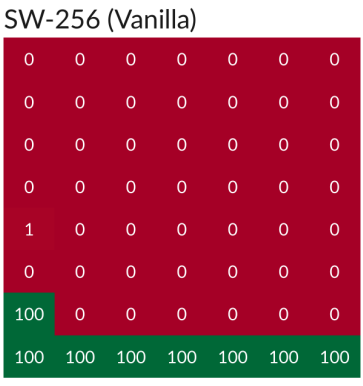
Long-context gains are strongest for the local backbone

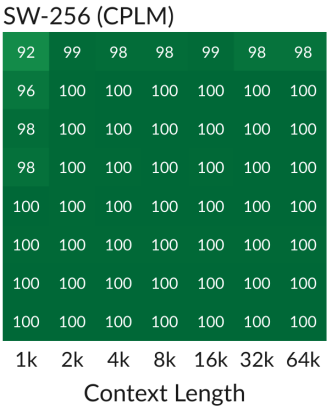
1.7B base-model score	Full V → CP	Hybrid V → CP	SW V → CP
BABILong	26.4 → 32.0	27.7 → 32.9	13.8 → 16.1
RULER	33.6 → 39.6	33.2 → 37.8	7.9 → 29.2
LongBench	10.4 → 12.2	9.5 → 14.3	7.4 → 13.4

- RULER: mean of 13 tasks across 4k–16k; BABILong: qa1–qa5 across 0k–16k.
- LongBench: mean over 16 English tasks at 16k.
- The small SW BABILong gain shows that long-range copying alone does not solve multi-fact reasoning.

Paper, Sec. 3.3, Table 3, PDF p. 6. Full/Hybrid use YaRN $\times 4$; SW uses base RoPE.

A 256-token-window model retrieves needles through 64k



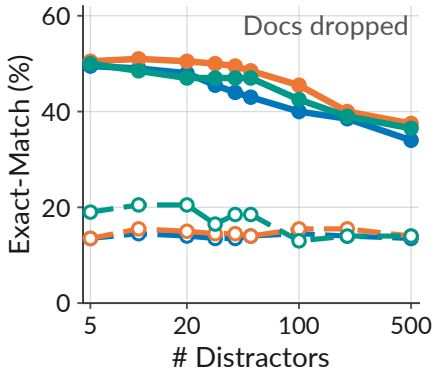


Grid mean
14.3 → 99.4%

Protocol
7-digit needle; 200 examples per cell; exact number in output.

Paper, Appendix E, Fig. 13, PDF p. 25. Cropped original SW panels; columns are 1k–64k context, rows are needle depths.

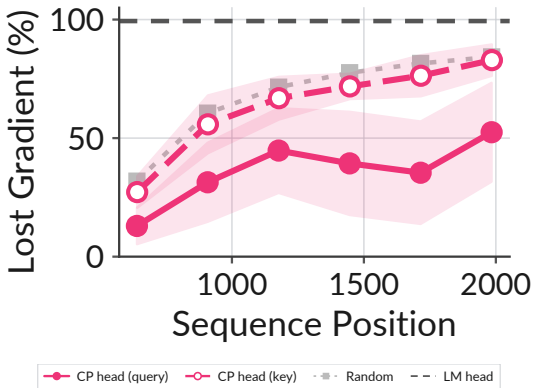
Copy keys can serve as compact document memory



- HotpotQA with documents hidden from answer-side backbone attention.
- CP keys alone give **34–38% EM** at 500 distractors.
- Reported cache is about **50× smaller** than Full document KV.
- Last-layer KV control stays near 14% EM.

Paper, Sec. 3.4, Fig. 5 (middle), PDF pp. 7–8. 200 validation questions; each regime is fine-tuned separately.

The CP path has a more favorable gradient shape



- LM head: over 99% of logit-gradient norm is lost in the measured setup.
- CP projections use t context positions rather than V vocabulary items; at early positions, up to 85% of gradient norm is retained.
- SpamLang: at $V = 512$k, vanilla reaches 34% accuracy; CPLM solves the copying task across tested V.

Paper, Sec. 4.1, Fig. 6, PDF pp. 8–9. Plot: lost gradient fraction vs. sequence position.

Two interventions test whether the gradient path matters

135M token model	Val. NLL ↓	PPL ↓
Vanilla	2.794	16.34
CPLM	**2.763**	**15.85**
CPLM, stop CP → backbone gradient	2.786	16.22

- Stop-gradient leaves 0.008 of the 0.031 nat/token gain: about one quarter.
- Byte-level $V = 257 < D = 576$: vanilla NLL 0.6908, CPLM 0.6918; no benefit when this LM-head rank bottleneck is removed.
- These targeted tests support the authors' gradient-path explanation.

Paper, Appendix H, Tables 15–16, PDF pp. 30–31. Byte NLL is per byte and is not comparable to token NLL.

The backbone delegates some copying to the CP head

- On a repeated random-sequence probe, CPLMs predict the repeated half more accurately.
- At sequence length $T = 128$, Full/Hybrid CPLM backbones contain **4** detected induction heads versus **13–15** in vanilla backbones.
- Low-copy-gate tokens also receive lower NLL in CPLMs, consistent with a better-trained shared backbone.

Interpretation

The CP head can carry copying work that vanilla models spread over backbone attention heads; the effect reaches the LM branch as well.

Paper, Sec. 4.2, Fig. 7, PDF p. 9; Appendix G.1, Fig. 14, PDF p. 27.

Measured overhead is small, with clear accounting

Resource	Reported measurement
Parameters	Two $D \times D$ projections + sink; about +0.5% at 1.7B.
Training	1.7B: 55,862 $\rightarrow$ 54,113 tokens/s on one B300 at 4k ($0.969\times$).
Inference	1.7B Full: copy cache = 2.1% of KV cache; decoding step about 1.7–1.8% slower in reported settings.

- 9B and 30B results in the appendix measure overhead on *untrained* architectures, not task quality.
- Fused and chunked training kernels control temporary memory.

Paper, Appendix B, Tables 5–6, PDF p. 19.

What the evidence establishes—and where it stops

1. A trained pointer head improves LM pretraining across the tested 135M–1.7B sizes and Full, SW, Hybrid and GDN backbones.
2. Gradient interventions explain much of the gain; the pointer design itself is related to prior copy mechanisms.
3. SW gains are striking for retrieval, but complex multi-fact tasks remain hard; Full models still fail the 16k+ needle test.

Take-home message

Giving the model an explicit way to copy changes both what it can access and how the backbone is trained.

Paper, Secs. 3–6 and Appendices E–H, PDF pp. 4–10, 25–32.

Paper and pointers for discussion

- Nathan Godey and Yoav Artzi. *Copy-Paste Language Models*. arXiv:2610.19247v1, 2026.
  https://arxiv.org/abs/2610.19247v1
- Main mechanism: Sec. 2, Eqs. (2)–(7), PDF p. 3.
- Evaluation: Sec. 3, Tables 1–3, PDF pp. 4–8.
- Mechanistic interventions: Appendix H, Tables 15–16, PDF pp. 30–32.

Discussion: when is a learned pointer a better use of memory than longer backbone attention?

Backup: pointer designs are close on small-scale perplexity

Method	Validation PPL Full	Validation PPL SW	FinePDF PPL Full	FinePDF PPL SW
Vanilla	16.34	16.88	14.25	16.52
Sigmoid gate	15.94	16.20	13.39	14.25
Shared softmax	15.88	16.09	13.32	**13.82**
CPLM + sink	**15.85**	**16.02**	**13.31**	**13.82**

- Margins among pointer variants are small; the paper emphasizes training with a pointer, not a novel pointer operator.

Paper, Appendix C.4, Table 8, PDF p. 23. 135M, 20B tokens, seed 42.

Backup: retrieval and cache comparisons

- A tuned inference-time neural cache barely changes 1.7B FineWeb-Edu PPL: Full 9.17 $\rightarrow$ 9.17, Hybrid 9.20 $\rightarrow$ 9.20, SW 9.65 $\rightarrow$ 9.63.
- Trained CPLMs score 9.06, 9.06 and 9.24, respectively, on the same 100-document evaluation.
- At 135M, reducing CP key/query dimension from 576 to 64 changes validation PPL by less than 0.02 while using 9$\times$ fewer copy-key values per token.

Paper, Appendix C.2 and C.5, Fig. 8 and Table 9, PDF pp. 20, 24.