SEMANTIC FUSION AND SUBSPACE SEPARATION IN LANGUAGE MODELS: A HYPOTHESIS AND ITS PREDICTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a hypothesis of semantic fusion and subspace separation in language models. Early information exchange may build shared representations of semantic units, reducing independent constituent access, while explicit boundaries may favor separately readable subspaces. The hypothesis predicts first-constituent substitutions in predecessor retrieval and their reduction with separators. Across three instruction-tuned language models in English and Chinese, semantic pairs elicit more such errors than random triples, and commas reduce them on paired items. To study the proposed mechanism, we construct a minimal arithmetic-and-retrieval task that permits explicit control over pair meanings and retrieval exposure. A small Transformer exhibits the predicted substitutions for operator pairs assigned arithmetic meanings but withheld from retrieval training. Blocking the early first-to-second-token attention pathway improves retrieval while sharply impairing arithmetic. These findings support the hypothesis's predictions; the proposed subspace organization remains a mechanistic interpretation.

1 INTRODUCTION

Language models encode multi-token lexical units (Elhage et al., 2022a; Kaplan et al., 2025) and compound meanings (Ormerod et al., 2024), while probes reveal reduced token recoverability at multi-token word and entity endings (Feucht et al., 2024). How does semantic organization affect constituent access, and can explicit boundaries change it? In "ice cream lake," "ice cream" has a composite meaning, yet "cream" remains the predecessor of "lake."

Studies of superposition (Elhage et al., 2022b), linear representations (Park et al., 2024), and contextual binding (Feng & Steinhardt, 2024) connect representational geometry to interference and access. Induction heads provide an attention pathway for token retrieval (Olsson et al., 2022).

We hypothesize a link between *semantic fusion* and *subspace separation*. For a semantic pair, early exchange builds a shared representation that can dominate constituent readout. Commas are hypothesized to favor distinct subspaces that preserve independent access.

Given a, b, Q, strict predecessor retrieval requires only b, without reasoning. Reading out the unit ab from its beginning can yield a or ab: "ice" or "ice cream" for "ice cream lake." Separators should reduce this error; disrupting a fusion pathway should improve constituent access while impairing computation using the pair's meaning.

In our minimal toy Transformer, three pairs learn arithmetic meanings but are withheld from retrieval training. Across five seeds, arithmetic and standard retrieval average 99.73% and 99.00% accuracy, yet held-out pairs elicit first-token errors. Blocking first-to-second-token attention in layer 0 improves retrieval while impairing arithmetic.

Three instruction-tuned LLMs show both behavioral predictions in English and Chinese: semantic pairs elicit more first-constituent errors than independently sampled random triples, and commas reduce these errors on paired items.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2 Related Work

Prior work studies token representations, phrasal composition, and compound semantics (Voita et al., 2019; Yu & Ettinger, 2020; Ormerod et al., 2024). Elhage et al. (2022a) describe early neurons responding to multi-token words, names, and compound expressions, motivating a detokenization account. Gurnee et al. (2023) identify compound-sensitive neurons through sparse probing. Lad et al. (2024) relate early token integration to local attention and sensitivity to layer interventions. Kaplan et al. (2025) study how multiple subword tokens form whole-word representations, examining early aggregation and FFN updates and reusing the resulting representations for vocabulary expansion. Our English probe instead examines semantic phrases composed of words that each occupy one token. The retrieval difficulty is particularly striking because the target is already a complete, single-token word at input.

Feucht et al. (2024) provide a close precedent: linear probes reveal reduced recoverability of current and preceding token identities at the final positions of multi-token words and entities. They interpret this "token erasure" as a footprint of lexical representations and use it to identify implicit vocabulary items. These findings motivate testable predictions about how semantic organization affects constituent access.

Our contribution combines predictions for frontier LLM behavior with an adjustable training setup. The hypothesis predicts more first-constituent substitutions for semantic pairs and fewer after inserting separators. Both directions hold across three LLMs in English and Chinese, including Kimi K3 with 2.8T total parameters (Kimi Team, 2026). In our minimal joint task, arithmetic assigns verifiable pair meanings and retrieval tests constituent access. Pair meanings, retrieval exposure, and task balance can be varied explicitly, enabling further study of how the proposed mechanism develops. In the reported setup, pairs are withheld from retrieval training; the same early attention knockout (Geva et al., 2023) improves retrieval while impairing arithmetic.

3 Semantic Fusion and Subspace Separation

Induction heads illustrate how information in different subspaces can be read out separately (Elhage et al., 2021; Olsson et al., 2022). A characteristic pattern is

$$[a][b] \ \cdots \ [a] \longrightarrow [b]. \tag{1}$$

Let e_a, e_b be column-vector token representations. An early previous-token head writes a transformed copy of e_a into the b position, giving a schematic state $W_O W_V e_a + e_b$. A later induction head's key projection reads the subspace carrying a: query–key (QK) matching locates the b position, and the output–value (OV) circuit copies b. Separately readable components allow matching on a while retrieving b.

We hypothesize that familiar pairs such as "ice cream" acquire a shared semantic representation at the second position, which can dominate readout over individual constituents. Explicit boundaries or low semantic relatedness are hypothesized to favor subspace separation: separately readable constituent representations, without requiring exact orthogonality. Figure 1 illustrates this distinction.

Let $x_i = a$ and $x_{i+1} = b$ be adjacent tokens forming the semantic unit ab. Write $h_j^{(\ell)} \in \mathbb{R}^{d_{\text{model}}}$ for the residual-stream representation at position j and layer ℓ, and $H^{(\ell)}$ for the representations at all positions. We hypothesize that fusion begins at an early layer ℓ_0, when an attention head m transfers a non-negligible contribution from a to b:

$$\Delta h_{b\leftarrow a}^{(\ell_0,m)} = \alpha_{i+1,i}^{(\ell_0,m)} W_O^{(\ell_0,m)} W_V^{(\ell_0,m)} \,\mathrm{LN}\Big(h_i^{(\ell_0)}\Big), \qquad \Big\|\Delta h_{b\leftarrow a}^{(\ell_0,m)}\Big\|_2 \geq \tau_\Delta > 0, \tag{2}$$

Here, $\alpha_{i+1,i}^{(\ell_0,m)}$ is the attention weight from the position of b to that of a; W_V and W_O are the value and output projections; and LN denotes layer normalization.

Information transfer alone does not require the receiving token to immediately encode the complete meaning of ab. Let

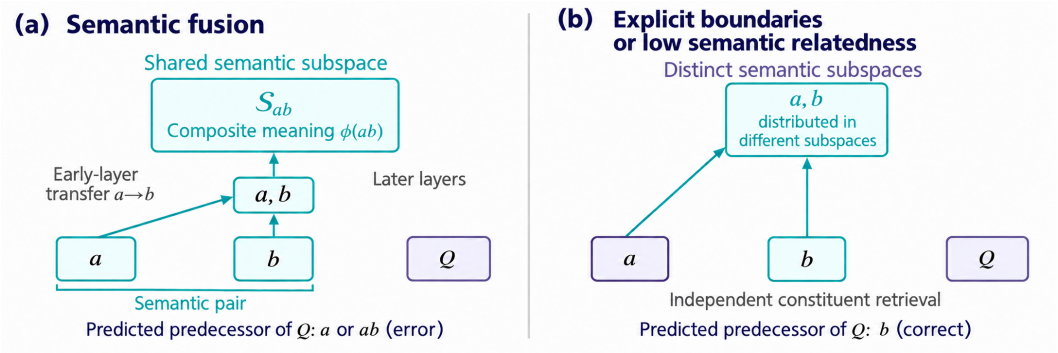

Figure 1: The fusion and separation hypothesis. Meaningful pairs are hypothesized to acquire shared semantic representations; explicit boundaries or low semantic relatedness favor separately readable constituents. The subspaces shown are schematic.

$$H^{(\ell+1)} = F^{(\ell)}\left(H^{(\ell)}\right) \tag{3}$$

denote the transformation performed by a full Transformer block, including attention and MLP computations. The semantic-fusion hypothesis posits that the representation at b encodes the composite meaning at some later layer $L > \ell_0$:

$$\exists L > \ell_0 \quad \text{such that} \quad \Pi_{\mathcal{S}_{ab}} h_{i+1}^{(L)} \approx \phi(ab), \tag{4}$$

The shared subspace $\mathcal{S}_{ab}$ is associated with the semantic unit ab, $\Pi_{\mathcal{S}_{ab}}$ is its projection, and $\phi(ab)$ is a reference representation of the composite semantics. The layer L may be $\ell_0 + 1$ or any subsequent layer; the hypothesis does not require fusion to extend across multiple layers.

Prediction 1: Constituent substitution. If the representation at b is accessed as the unit ab and decoded from its beginning, a predecessor query can yield a or ab instead of b. This readout assumption predicts more first-constituent errors for semantic than random abQ sequences.

Prediction 2: Explicit separation. Inserting commas between the same constituents is hypothesized to favor distinct subspaces, improving independent access to b and reducing the first-constituent error. The punctuation comparison tests the behavioral consequence of this proposed representational change.

Prediction 3: Pathway intervention. Blocking an early pathway supporting fusion should improve constituent retrieval while impairing a computation that uses the pair's meaning. The joint arithmetic-and-retrieval task tests these opposing effects in the same model.

4 Controlled Toy Model

The toy setting tests the substitution and pathway predictions under known training conditions. Arithmetic assigns verifiable meanings to adjacent token pairs, while predecessor retrieval measures access to their constituents. We train one decoder-only Transformer (Vaswani et al., 2017) on both tasks, excluding the designated pairs from retrieval training.

4.1 Model Architecture

The model uses rotary position embeddings (RoPE; Su et al., 2021) and pre-LayerNorm inputs (Xiong et al., 2020) $\hat{h}_i^{(\ell)} = \mathrm{LN}(h_i^{(\ell)})$. For head m in layer ℓ, queries, keys, and values are computed as follows, omitting affine projection biases:

$$q_i^{(\ell,m)} = \mathrm{RoPE}\Big(W_Q^{(\ell,m)}\hat{h}_i^{(\ell)}, i\Big), \quad k_i^{(\ell,m)} = \mathrm{RoPE}\Big(W_K^{(\ell,m)}\hat{h}_i^{(\ell)}, i\Big), \quad v_i^{(\ell,m)} = W_V^{(\ell,m)}\hat{h}_i^{(\ell)}. \tag{5}$$

The causal self-attention output at position i is

$$\eta_i^{(\ell,m)} = \sum_{j\le i} \mathrm{softmax}_j\left(\frac{{q_i^{(\ell,m)}}^\top k_j^{(\ell,m)}}{\sqrt{d_h}}\right) v_j^{(\ell,m)}, \tag{6}$$

where d_h is the head dimension. The outputs of all M heads are concatenated and projected:

$$o_i^{(\ell)} = W_O^{(\ell)}\,\mathrm{Concat}\Big(\eta_i^{(\ell,1)}, \ldots, \eta_i^{(\ell,M)}\Big). \tag{7}$$

The head-specific $W_O^{(\ell,m)}$ is the corresponding column block of $W_O^{(\ell)}$. Let $O^{(\ell)}$ collect the projected outputs at all positions. Residual connections and layer normalization give

$$\bar{H}^{(\ell)} = H^{(\ell)} + O^{(\ell)}, \qquad H^{(\ell+1)} = \bar{H}^{(\ell)} + \mathrm{MLP}\Big(\mathrm{LN}(\bar{H}^{(\ell)})\Big). \tag{8}$$

The Transformer hyperparameters are in Appendix A. For the architecture comparison, we also use a four-layer hybrid with kernel linear attention and width-3 causal depthwise convolution in its first and third layers, and softmax attention in its second and fourth layers. Appendix D describes the hybrid implementation and its hyperparameters.

4.2 Vocabulary and Task Definitions

Table 1 defines the vocabulary. The 331 value tokens represent operands and results; tokens 0, 1, and 2 are also used in the three adjacent pairs that encode arithmetic operations (Table 2). These pairwise meanings supply the controlled semantic units for the retrieval experiments.

Table 1: Token dictionary for the toy model.

Token ID	Description
0–330	331 value tokens; tokens 0, 1, and 2 are additionally reused to encode arithmetic operators.
331	`QUERY`: indicates that the retrieval target token follows.
332	`SEP`: separates the memory sequence from the query.
333	`BOS`: marks the beginning of a sequence.
334	`EOS`: marks the end of a sequence.
335	`PAD`: pads sequences to a uniform length.

Table 2: Arithmetic semantics encoded by adjacent token pairs, with integer operands r, s. The typewriter names identify task conditions.

Name	Adjacent Tokens	Arithmetic Semantics
`q01/add`	$0, 1$	$(r + s) \bmod 331$
`q02/sub`	$0, 2$	$(r - s) \bmod 331$
`q21/sub1`	$2, 1$	$(r - s - 1) \bmod 331$

Arithmetic. The input is $[\texttt{BOS}, r, \mathrm{op}_1, \mathrm{op}_2, s]$, where r, s are operands and $(\mathrm{op}_1, \mathrm{op}_2)$ specifies one of the three operations. The model must output the result modulo 331. Figure 2 illustrates addition: the expression "10 + 15" has the expected output "25."

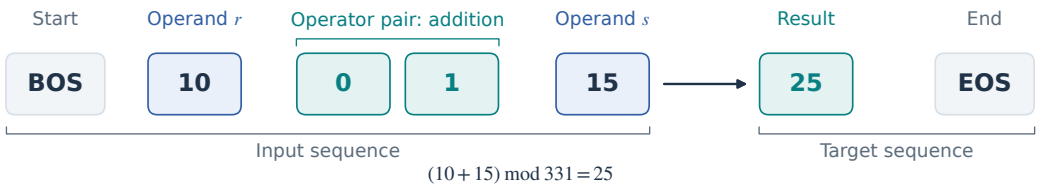

Figure 2: An example of the arithmetic task.

Predecessor retrieval. The input sequence consists of a memory followed by a query: $[\texttt{BOS}, x_1, x_2, \ldots, x_n, \texttt{SEP}, \texttt{QUERY}, Q]$, with $3 \leq n \leq 6$. The model must find the query token Q in the memory and output the token immediately before it. In Figure 3, the memory is "16, 134, 27, 227," the query target is "227," and the correct answer is "27."

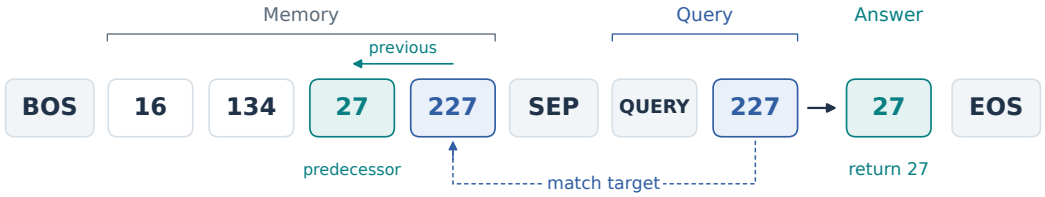

Figure 3: An example of the retrieval task.

4.3 Training Data

The training set contains 270,000 arithmetic examples and 135,000 retrieval examples. The arithmetic examples satisfy two constraints:

- For each operation, 90,000 operand pairs are sampled without replacement from the remaining operand pairs.
- For each operation, the left operand, right operand, and result each cover all 331 value tokens.

Retrieval examples are constructed with the following constraints:

- Each memory contains unique tokens drawn from 0–330.
- The first memory token is never selected as the target because it has no predecessor.
- Each problem signature, `(memory, target_index)`, is unique.
- No retrieval-training memory contains the adjacent pair $(0, 1)$, $(0, 2)$, or $(2, 1)$.
- The ordered pair `(query_target, answer)` is likewise never $(0, 1)$, $(0, 2)$, or $(2, 1)$.

The designated pairs therefore acquire arithmetic meanings without direct supervision on retrieving their constituents.

4.4 Evaluation Sets

We evaluate three complementary test subsets:

- **Arithmetic:** 12,288 examples, with 4,096 per operation.
- **Standard retrieval:** 2,000 predecessor-retrieval examples without arithmetic operators, measuring generalization on the ordinary task.
- **Retrieval with arithmetic operators:** 7,680 examples whose memories contain an operator pair (a, b) together with a query token Q. The model must return the token immediately preceding Q.

5 TOY-MODEL RESULTS

We first establish the predicted retrieval error and check whether it can be explained by the identity of the correct answer token. We then intervene on the proposed attention pathway to examine its role in arithmetic composition and predecessor retrieval, and compare with a hybrid architecture to examine whether composition can coexist with better constituent access.

5.1 RETRIEVAL ERRORS ON SEMANTIC PAIRS

Table 3 reports Transformer results for seeds 101–105. For each designated operator pair, **First** denotes an incorrect first-token response, **Correct** the true predecessor, and **EOS** an incorrect end-of-sequence output. These columns report the three specified outcomes, rather than an exhaustive partition of all outputs.

Table 3: Arithmetic and retrieval test results (all values in %).

Seed	Arithmetic Accuracy	Standard Retrieval Acc.	$(0,1)$			$(0,2)$			$(2,1)$		
			First	Correct	EOS	First	Correct	EOS	First	Correct	EOS
101	99.63	99.70	31.33	0.20	62.46	30.39	22.70	31.91	54.10	0.70	33.16
102	100.00	98.50	46.45	17.03	3.32	68.67	12.93	12.11	70.39	9.45	12.34
103	99.93	100.00	17.30	5.70	0.00	21.29	44.06	0.04	70.55	3.16	0.00
104	99.68	98.30	39.22	37.70	19.22	22.66	60.94	11.02	46.56	35.82	16.25
105	99.42	98.50	59.02	12.11	0.35	38.75	29.26	0.27	84.41	5.74	0.16
Mean	**99.73**	**99.00**	**38.66**	**14.55**	**17.07**	**36.35**	**33.98**	**11.07**	**65.20**	**10.98**	**12.38**

The model achieves mean accuracies of 99.73% on arithmetic and 99.00% on standard retrieval. Performance drops substantially when retrieval involves the semantic pairs: mean correct-output rates are 14.55%, 33.98%, and 10.98% for $(0,1)$, $(0,2)$, and $(2,1)$, respectively. The corresponding first-token error rates are 38.66%, 36.35%, and 65.20%.

Thus, the model can retrieve the predecessor of Q in abQ when ab is not a designated semantic unit, but often returns a when ab is an arithmetic operator pair. This pattern supports the proposed account in which the model processes the learned pair holistically and retrieves its first token instead of the actual predecessor. Some runs also emit `EOS`; one possible explanation is that arithmetic training sequences are shorter than retrieval sequences.

To test whether the error simply reflects difficulty retrieving tokens 1 and 2, we evaluate control sequences $x\,1\,y$ and $x\,2\,y$, querying y. Here x, y are control tokens, with x chosen so that neither $(x,1)$ nor $(x,2)$ is a designated operator pair. As Table 4 shows, mean correct-output rates remain 92.81% and 93.82%, respectively. These controls rule out the simple explanation that retrieval fails merely because the predecessor is token 1 or token 2.

Table 4: Correct-output rates for the $x\,1\,y$ and $x\,2\,y$ control sequences across random seeds.

Seed	Correct Output ($x\,1\,y$)	Correct Output ($x\,2\,y$)
101	90.70%	94.14%
102	91.91%	94.80%
103	99.69%	97.73%
104	87.11%	89.80%
105	94.65%	92.62%
Mean (101–105)	**92.81%**	**93.82%**

5.2 ABLATING THE PROPOSED FUSION PATHWAY

The fusion hypothesis predicts a coupled effect of disrupting the pathway from the first token to the second: retrieving the second token should become easier, while performing the operation encoded

by the pair should become harder. We test this prediction using attention-edge masking, as in attention knockout (Geva et al., 2023): across all heads in layer 0, we block the position of b from attending to the position of a, where a, b are the two tokens of the semantic pair.

Table 5: Layer-0 attention-pathway ablation. Entries show baseline → masked accuracy; pp denotes percentage points.

Seed	q01 Accuracy	q02 Accuracy	q21 Accuracy	Arithmetic Held: Baseline → Masked
101	0.20% → **66.64%**	22.70% → **57.62%**	0.70% → **58.87%**	99.63% → **2.12%** (−97.51 pp)
102	17.03% → **63.01%**	12.93% → **60.82%**	9.45% → **63.16%**	100.00% → **14.74%** (−85.26 pp)
103	5.70% → **93.12%**	44.06% → **80.82%**	3.16% → **87.89%**	99.93% → **4.13%** (−95.80 pp)
104	37.70% → **70.23%**	60.94% → **90.78%**	35.82% → **77.30%**	99.68% → **19.89%** (−79.79 pp)
105	12.11% → **89.96%**	29.26% → **91.64%**	5.74% → **95.27%**	99.42% → **13.03%** (−86.39 pp)
Average	14.55% → **76.59%**	33.98% → **76.34%**	10.98% → **76.50%**	99.73% → **10.78%** (−88.95 pp)

Table 5 shows improved retrieval for all three pairs and all five seeds. Mean retrieval accuracy rises from 14.55% to 76.59% for `q01`, from 33.98% to 76.34% for `q02`, and from 10.98% to 76.50% for `q21`. At the same time, arithmetic accuracy falls from 99.73% to 10.78%, a decline of 88.95 percentage points. The opposing effects connect the early attention pathway to both the useful composition of operator meanings and the interference observed in constituent retrieval.

5.3 Constituent Retrieval in a Hybrid Model

The pathway ablation examines how the Transformer uses early attention. The hybrid comparison asks whether a different architecture can retain arithmetic composition while making the constituents easier to retrieve. In the reported five-seed summary (Table 13), mean semantic-pair retrieval accuracy is 79.65% for the hybrid versus 19.84% for the original Transformer reference, while first-constituent errors are 10.67% versus 46.74%. Arithmetic accuracy remains near perfect (99.99% versus 99.73%); standard retrieval accuracy is 97.70% versus 99.00%. These results suggest that architecture can affect constituent accessibility while preserving strong compositional performance, consistent with semantic fusion coexisting with additional routes to individual tokens. Short convolutions may help retain constituent information alongside the composite representation, contributing to the improved retrieval. Appendix D reports the individual runs and the evaluation epochs included in the summary.

6 Behavioral Predictions in Large Language Models

The toy experiment examines the proposed early pathway. We next test the substitution and separation predictions through generated answers from three API models: Kimi K3, DeepSeek V4 Pro, and Hy4 preview. Given a, b, Q, the model must identify the constituent immediately before Q, which is always b. We compare semantic and random triples, then insert commas into the same semantic triples to test whether explicit boundaries reduce the predicted first-constituent error.

6.1 Task and Experimental Design

We evaluate three conditions in English and Chinese, with 500 items per condition shared across Kimi K3, DeepSeek V4 Pro, and Hy4 preview. This yields 3,000 distinct prompts and 9,000 completed responses. Each item contains three distinct English words or Chinese characters.

In the *Semantic* condition, ab forms a familiar unit: for example, "ice cream lake" queries "lake" and has the answer "cream." We use 25 English expressions with 20 query words each, and 118 Chinese pairs with four or five query characters each. English words are separated by spaces; Chinese characters are contiguous. The full abQ sequence need not be meaningful. The *Random* condition independently draws three distinct constituents uniformly from each language's candidate pool, resampling every position rather than shuffling semantic items. The *Comma* condition uses exactly the same semantic triples, with comma-plus-space separators inserted between constituents. Only the separators change in this paired comparison.

Table 6: Outcomes for the first proposed answer in English. Each row contains 500 completed trials. All entries are percentages; the four outcomes sum to 100%. Semantic and Comma use the same triples. Bold marks the highest accuracy within each model.

Model	Condition	Correct b	Wrong a	Wrong Q	Other
Kimi K3	Semantic	83.8	12.6	0.0	3.6
	Random	94.4	5.0	0.0	0.6
	Comma	**100.0**	0.0	0.0	0.0
DeepSeek V4 Pro	Semantic	33.6	32.8	0.2	33.4
	Random	**75.0**	11.4	1.4	12.2
	Comma	55.2	12.0	2.8	30.0
Hy4 preview	Semantic	65.8	30.8	0.0	3.4
	Random	89.0	7.8	0.2	3.0
	Comma	**93.2**	6.0	0.0	0.8

Table 7: Outcomes for the first proposed answer in Chinese. Each row contains 500 completed trials. All entries are percentages; the four outcomes sum to 100%. Semantic and Comma use the same triples. Bold marks the highest accuracy within each model.

Model	Condition	Correct b	Wrong a	Wrong Q	Other
Kimi K3	Semantic	76.4	23.4	0.2	0.0
	Random	97.4	2.6	0.0	0.0
	Comma	**100.0**	0.0	0.0	0.0
DeepSeek V4 Pro	Semantic	31.8	67.4	0.0	0.8
	Random	66.0	34.0	0.0	0.0
	Comma	**94.2**	5.6	0.0	0.2
Hy4 preview	Semantic	55.6	40.8	0.4	3.2
	Random	91.0	8.8	0.2	0.0
	Comma	**99.8**	0.0	0.0	0.2

Every unpunctuated sequence contains exactly three aligned tokens under all three models' cached official tokenizers, including at the start of the actual prompt. Commas add separator tokens; neither the comma sequence nor the full question is constrained to three tokens. Each request uses one user message, no system prompt or history, temperature 0, a 32-token output limit, and disabled reasoning. The prompt asks for only one word or character. Appendix B gives the candidate pools, tokenization checks, literal prompts, and provider-specific settings.

6.2 Scoring the First Answer

We assess whether the model's first actual answer identifies b, without requiring b to be the first generated token. Introductory text, formatting, and restatements of the question do not count as answers. Thus, "The word is cream" is correct when the gold answer is "cream." In contrast, "ice. Actually, cream" is judged by "ice": a later correction does not replace an earlier answer. The four exclusive outcomes are *Correct* b, *Wrong* a, *Wrong* Q, and *Other*. Other includes answers outside the triple and responses without an answer. Percentages use all 500 completed items per condition. Appendix B.2 gives the detailed rules and examples.

6.3 Results Across Models and Languages

Semantic pairs are harder than random triples. Tables 6 and 7 show the same direction of change for every model in both languages: semantic items have lower accuracy and more first-constituent errors than random items. For example, DeepSeek's Chinese accuracy is 31.8% on semantic items versus 66.0% on random items, while its Wrong a rate rises from 34.0% to 67.4%. Hy4's English accuracy is 65.8% versus 89.0%, with Wrong a rising from 7.8% to 30.8%. Kimi is

more accurate overall but shows the same pattern. The random baseline itself is not error-free, so returning a is not unique to a deliberately constructed semantic pair.

Model differences and architecture. Kimi K3 achieves 83.8% accuracy on English semantic pairs and 76.4% on Chinese semantic pairs, substantially above the other two models. Its hybrid architecture offers one possible explanation: Kimi Delta Attention, which includes short convolutions, is interleaved with global attention (Kimi Team, 2026). Convolutional processing may provide additional routes for retaining constituent information alongside the composite representation of ab (Arora et al., 2024). The hybrid toy-model results in Section 5.3 illustrate how strong composition can coexist with better constituent access, although the specific mechanism in Kimi remains untested.

Commas improve retrieval, with model-dependent residual errors. On the paired English items, accuracy rises from 83.8% to 100.0% for Kimi, from 33.6% to 55.2% for DeepSeek, and from 65.8% to 93.2% for Hy4. On Chinese items, the corresponding changes are 76.4% to 100.0%, 31.8% to 94.2%, and 55.6% to 99.8%. First-constituent errors decrease in all six comparisons. Commas eliminate this error for Kimi in both languages and for Hy4 in Chinese, but do not solve every model's retrieval problem.

DeepSeek's English errors require a separate explanation. For DeepSeek, English Wrong a falls from 32.8% to 12.0% after adding commas, whereas Other falls only from 33.4% to 30.0%. Wrong Q increases from 0.2% to 2.8%; together, errors outside a remain almost unchanged at 33.6% versus 32.8%. Thus, the modest final accuracy is compatible with a large reduction in first-constituent errors: another class of error remains.

6.4 Consistency with the Theoretical Predictions

Across all three models and both languages, semantic pairs increase responses of a instead of b compared with random triples; commas reduce this same error on the paired items. These findings support the substitution and separation predictions of the hypothesis at the behavioral level. Appendix C provides additional error examples.

7 Discussion and Conclusion

From the hypothesis to its tests. The hypothesis links semantic grouping and explicit boundaries to constituent accessibility. The LLM results support the predicted substitution and boundary effects, although individual errors may also reflect instruction interpretation, lexical familiarity, or positional retrieval. The proposed subspace organization remains a mechanistic interpretation rather than a directly measured result.

A possible role of chain-of-thought formatting. Chain-of-thought prompting elicits intermediate reasoning steps (Wei et al., 2022); scratchpad training likewise uses generated intermediate computations (Nye et al., 2021). In predecessor retrieval, a model given an unpunctuated sequence may restate it as "a, b, Q" before correctly identifying b. Under our hypothesis, these boundaries favor distinct semantic subspaces. The rewritten text becomes context for subsequent generation, allowing retrieval from the separated representation. Likewise, reasoning organized as "1. … 2. … 3. …" gives facts or steps explicit boundaries. One possible benefit of chain-of-thought is thus to reformulate information so that different meanings remain separately accessible in latent space.

The toy experiment supports the pathway prediction: blocking early attention improves constituent retrieval while impairing arithmetic, connecting useful composition and retrieval interference to the same pathway. Together with the LLM boundary effects, this supports further investigation of semantic fusion and subspace separation as a joint account of constituent access.

AI USE STATEMENT

Generative AI tools, including Codex and ChatGPT, were used in this work. Specifically, GPT-6 was used to generate the training code for both the standard Transformer and hybrid toy models. GPT-6 was also used to generate the prompts for the LLM experiments, construct random triples, and select English word pairs and Chinese character pairs with strong semantic associations. Additional uses included assistance with synthetic dataset generation, processing model responses, feedback on experimental procedures, literature searches, language polishing, manuscript organization, and figure editing. We take responsibility for the final text, code, analyses, and claims.

ETHICS STATEMENT

The experiments use synthetic arithmetic and retrieval sequences and constructed word/character prompts. They involve no human participants or collection of personal data and are intended to study model representations and retrieval behavior.

REPRODUCIBILITY STATEMENT

Sections 4 and 5 describe the toy tasks, data-generation constraints, evaluation sets, and attention-pathway intervention. Appendix A lists the Transformer hyperparameters, and Appendix D specifies the hybrid architecture, training configuration, per-seed results, and evaluation snapshots used in the reported summary. Section 6 and Appendix B document the LLM benchmark conditions, sampling procedures, tokenization checks, prompt templates, API generation settings, and scoring rules. The anonymous supplementary package provides toy-model training implementations and the LLM prompts, evaluation runner, and scoring code; its README documents the provenance and scope of the supplied implementations.

REFERENCES

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall-throughput tradeoff. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 1763–1840. PMLR, 2024. URL https://proceedings.mlr.press/v235/arora24a.html.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Nelson Elhage, Tristan Hume, Catherine Olsson, Neel Nanda, Tom Henighan, Scott Johnston, Sheer ElShowk, Nicholas Joseph, Nova DasSarma, Ben Mann, Danny Hernandez, Amanda Askell, Kamal Ndousse, Andy Jones, Dawn Drain, Anna Chen, Yuntao Bai, Deep Ganguli, Liane Lovitt, Zac Hatfield-Dodds, Jackson Kernion, Tom Conerly, Shauna Kravec, Stanislav Fort, Saurav Kadavath, Josh Jacobson, Eli Tran-Johnson, Jared Kaplan, Jack Clark, Tom Brown, Sam McCandlish, Dario Amodei, and Christopher Olah. Softmax linear units. *Transformer Circuits Thread*, 2022a. URL https://transformer-circuits.pub/2022/solu/index.html.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022b. URL https://transformer-circuits.pub/2022/toy_model/index.html.

Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/9d1b7fc578c0d2d6431fc26d736ecaf3-Abstract-Conference.html.

Sheridan Feucht, David Atkinson, Byron C. Wallace, and David Bau. Token erasure as a footprint of implicit vocabulary items in LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9727–9739, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.543. URL https://aclanthology.org/2024.emnlp-main.543/.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12216–12235, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.751. URL https://aclanthology.org/2023.emnlp-main.751/.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610*, 2023. URL https://arxiv.org/abs/2305.01610.

Guy Kaplan, Matanel Oren, Yuval Reif, and Roy Schwartz. From tokens to words: On the inner lexicon of LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/aef75887979ae1287b5deb54a1e3cbda-Abstract-Conference.html.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5156–5165, 2020. URL https://proceedings.mlr.press/v119/katharopoulos20a.html.

Kimi Team. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026. URL https://arxiv.org/abs/2607.24653.

Vedang Lad, Wes Gurnee, and Max Tegmark. The remarkable robustness of LLMs: Stages of inference? *arXiv preprint arXiv:2406.19384*, 2024. URL https://arxiv.org/abs/2406.19384v1.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *The Seventh International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

George A. Miller. WordNet: A lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995. doi: 10.1145/219717.219748. URL https://doi.org/10.1145/219717.219748.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021. URL https://arxiv.org/abs/2112.00114.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022. URL https://arxiv.org/abs/2209.11895.

Mark Ormerod, Jesús Martínez del Rincón, and Barry Devereux. How is a "kitchen chair" like a "farm horse"? exploring the representation of noun-noun compound semantics in transformer-based language models. *Computational Linguistics*, 50(1):49–81, 2024. doi: 10.1162/coli_a_00495. URL https://aclanthology.org/2024.cl-1.3/.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39643–39666. PMLR, 2024. URL https://proceedings.mlr.press/v235/park24c.html.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. URL https://arxiv.org/abs/2104.09864.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Elena Voita, Rico Sennrich, and Ivan Titov. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 4396–4406, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1448. URL https://aclanthology.org/D19-1448/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10524–10533. PMLR, 2020. URL https://proceedings.mlr.press/v119/xiong20b.html.

Lang Yu and Allyson Ettinger. Assessing phrasal representation and composition in Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language*

Processing (EMNLP), pp. 4896–4907, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.397. URL https://aclanthology.org/2020.emnlp-main.397/.

A Model and Training Hyperparameters

All five runs (seeds 101–105) use the model and training hyperparameters in Table 8. Only the random seed and the resulting training and evaluation examples differ across runs.

Table 8: Transformer hyperparameters used in all seed-101–105 runs.

Hyperparameter	Value
Training epochs	1,200
Batch size	8,190
Peak learning rate	1×10^{-3}
Learning-rate schedule	Linear warmup followed by cosine decay
Warmup duration	500 optimizer updates
Minimum learning-rate ratio	0.1 of the peak learning rate
Final learning rate	Approximately 1×10^{-4}
Vocabulary size	336
Maximum sequence length	12
Transformer blocks	4
Hidden dimension	256
Attention heads	8
Positional encoding	Interleaved RoPE on Q and K
RoPE base	10,000
MLP hidden dimension	1,024
Activation	GELU
Normalization	Pre-LayerNorm with final LayerNorm
Token embedding	336×256
Dropout	0
Parameter initialization	PyTorch defaults (no custom initialization)

Training objective. We use weighted causal next-token cross-entropy. For retrieval sequences, memory tokens, the query target, and the retrieval answer receive weights 0.1, 2, and 10, respectively. For arithmetic sequences, operands and the result receive weights 0.1 and 10, respectively. All other non-PAD targets have unit weight, and PAD targets are ignored.

Optimizer. We use AdamW (Loshchilov & Hutter, 2019) with $(\beta_1, \beta_2) = (0.9, 0.999)$ and weight decay 0.1.

B LLM Benchmark Construction and Evaluation Details

This appendix documents the evaluation in Section 6.

B.1 Dataset Construction and Prompts

We evaluate three conditions in each of two languages. Each condition contains 500 items, shared across all three models, giving 3,000 distinct prompts and 9,000 completed model responses. The constituents are English words or Chinese characters, and all three constituents within an item are distinct.

Semantic pairs. In English, ab is a familiar two-word expression, such as “ice cream,” “high school,” or “credit card.” We use 25 expressions, each paired with 20 different query words Q. Words are separated by spaces: for example, “ice cream lake” has the gold answer “cream.” In Chinese, we use 118 meaningful two-character pairs to construct 500 items, with four or five query characters per pair. Characters are written contiguously. For example, 猜中远 contains the pair 猜中 (“guess correctly”), and the answer is 中. Only ab is required to form a semantic unit; the full sequence abQ is not required to be a phrase or a word.

Independent random triples. We draw three distinct constituents uniformly without replacement within each triple, using fixed random seeds, and reject duplicate triples and items that fail the

tokenization check below. Each draw resamples all three positions; this condition does not shuffle the semantic items or hold their second and third constituents fixed. The Chinese pool contains 189 manually selected common characters. The English pool contains 2,694 lowercase alphabetic WordNet 3.0 entries (Miller, 1995) of at least two letters, filtered for single-token encoding both alone and after a space. Accidental semantic associations are not excluded. Both languages use the same sampling rule, although their pools differ in size, source, and lexical familiarity.

Comma-separated semantic pairs. We take the same 500 semantic triples in each language and insert an ASCII comma followed by a space between constituents: "ice cream lake" becomes "ice, cream, lake." The query and answer stay unchanged. This gives a paired comparison in which only the separators change. We do not add commas to the random condition.

Tokenization and prompts. For every unpunctuated triple, we verify that the displayed sequence consists of exactly three aligned tokens under the locally cached official tokenizers for all three models. Thus, each Chinese character occupies one token; in English, the tokens correspond to a, space-plus-b, and space-plus-Q. We also verify this alignment at the start of the actual user prompt. The three-token requirement applies to this initial sequence, not to the complete question. Commas introduce additional separator tokens, so the comma condition is not constrained to three tokens. Table 9 gives literal prompt examples. Semantic and random items use the same question and output instruction.

Table 9: Example user prompts and gold answers. The unpunctuated template is shared by semantic and random items; the comma template is used only for the paired semantic condition. The final column specifies the expected answer b, not an observed model response, and does not indicate whether a model answered correctly. Model performance is reported separately in Tables 6 and 7.

Language / format	User prompt	Gold answer
English / spaces	`ice cream lake. What word comes immediately before "lake"? Output only one word.`	cream
English / commas	`ice, cream, lake. What word comes immediately before "lake"? Output only one word.`	cream
Chinese / contiguous	猜中远。直接问"远"的前一个字是什么？只输出一个字即可。	中
Chinese / commas	猜, 中, 远。直接问"远"的前一个字是什么？只输出一个字即可。	中

Generation settings. Each request contains one user message, with no system message or conversation history. We set temperature to 0 and the output limit to 32 tokens, and disable reasoning through the provider-specific settings. Kimi K3 and DeepSeek V4 Pro use DashScope, with `reasoning_effort=none` and `enable_thinking=false`, respectively. Hy4 preview uses Tencent MaaS with `thinking={"type":"disabled"}`. No completed response returns reasoning text. Kimi and Hy4 report zero reasoning tokens; DeepSeek does not report that field. Only transport and rate-limit failures are retried; incorrect answers are retained.

B.2 Scoring the First Answer

Correctness depends on the first answer the model actually proposes, not on whether the response begins with the gold answer. We ignore introductory phrases, formatting, and complete restatements of the input or question when identifying that answer. For example, on "ice, hockey, spoon," the response "The word that comes immediately before 'spoon' is **hockey**" is correct: it identifies "hockey" as the predecessor. The introductory words are not competing answers. Similarly, "hot dog cake. The word immediately before 'cake' is dog" is correct because the opening triple restates the input and the first proposed answer is "dog."

An earlier proposed answer cannot be replaced by a later correction. For example, "ice. Actually, cream" is incorrect when the gold answer is "cream," whereas "The word is cream" is correct. An initially correct answer also remains correct if later text contradicts it. This evaluates the initial answer choice independently of explanatory wording.

The task requests one Chinese character or one English word. After isolating the answer, we normalize leading whitespace and, for English, punctuation and case while preserving internal apostrophes and hyphens. For a bare multiword proposal, error labeling follows its first returned constituent: "high school" on "high school fork" is a Wrong a response, because it returns the combined phrase instead of identifying "school" as the requested word. Merely including b in a phrase or a copy of the question is not enough to establish that the model selected b as its answer.

Responses are assigned to *Correct* b, *Wrong* a, *Wrong* Q, or *Other*. Other includes a proposed answer outside the input triple, refusals, question-only copies, and empty or otherwise unanswered responses. All percentages use the 500 completed items in the condition as their denominator.

C Additional English Error Examples

Table 10 shows five cases in which DeepSeek gives an incorrect answer. For "ice, cream, belt," the expected predecessor is "cream," but the model answers "seat." This answer forms the familiar expression "seat belt" with the query word, yet does not identify the predecessor in the displayed sequence. The other examples show the same pattern in the semantic, comma, and random conditions. These cases suggest that familiar expressions ending in Q can compete with the requested positional lookup.

Table 10: Examples of incorrect answers from DeepSeek V4 Pro, classified as Other. Gold b is the expected predecessor; Raw output shows the complete model response. The final column illustrates a familiar expression formed by the incorrect answer and the query word Q.

Condition	Input sequence	Gold b	Raw output	Output + Q
Semantic	post office wheel	office	Ferris	Ferris wheel
Semantic	science fiction rain	fiction	acid	acid rain
Comma	ice, cream, belt	cream	seat	seat belt
Comma	science, fiction, root	fiction	square	square root
Random	due pot pad	pot	launch	launch pad

D Hybrid Architecture and Additional Toy-Model Results

D.1 Architecture and Constituent Accessibility

Hybrid attention combines different token-mixing mechanisms within one network. Kimi K3 interleaves three Kimi Delta Attention (KDA) layers with one Gated Multi-head Latent Attention layer, and places another global-attention layer at the end of the backbone; its KDA computation includes short convolutions (Kimi Team, 2026). These components motivate an architectural explanation for its strong predecessor retrieval. Local convolutions may carry information about individual neighboring tokens alongside information about their combination, leaving a and b independently readable even when ab has acquired a shared meaning. This coexistence is compatible with the semantic-fusion hypothesis.

To explore this possibility in the toy setting, we use a simplified four-layer hybrid. Its first and third layers use kernel linear attention with a causal depthwise convolution; its second and fourth layers retain standard causal softmax attention. The model retains width 256, eight heads, GELU MLPs of width 1,024, pre-LayerNorm residual blocks, a final LayerNorm, and the 336-token vocabulary. The convolution has width 3 and acts separately on each channel of the projected queries, keys, and values, before RoPE is applied to queries and keys. For a projected query, key, or value sequence z, channel c is transformed as

$$\widetilde{z}_{i,c} = \beta_c^{\mathrm{conv}} + \sum_{t=0}^{2} w_{t,c} z_{i-t,c}, \tag{9}$$

where β_c^{conv} is the channel bias and t is the causal offset, with zero padding before the sequence starts. The separate coefficients for different offsets provide a possible route for preserving the identities of neighboring constituents.

The linear-attention layers use the positive feature map $\psi(x) = \mathrm{ELU}(x) + 1$ (Katharopoulos et al., 2020). Here, q_i, k_i denote queries and keys after convolution and RoPE, and $\widetilde{v}_i$ denotes convolved values. Their per-head output, with layer and head indices omitted, is

$$\eta_i = \frac{\sum_{j \le i} \psi(q_i)^\top \psi(k_j)\, \widetilde{v}_j}{\sum_{j \le i} \psi(q_i)^\top \psi(k_j)}. \tag{10}$$

The implementation evaluates this causal kernel directly for the short toy sequences. The hybrid has 3,338,064 parameters, including 6,144 convolution parameters. It tests the combination of linear attention and local convolution; it does not reproduce KDA's delta-rule memory or the full Kimi K3 architecture, and does not isolate convolution from the attention change.

D.2 Training and Reported Evaluations

Table 11 collects the model and training hyperparameters of the hybrid implementation, including the settings shared with the Transformer reference.

The hybrid uses the reconstructed toy task with 270,000 arithmetic and 135,000 retrieval training examples, and seeds 101–105. The designated operator pairs are excluded from retrieval training as described in Section 4. Evaluation includes 12,288 held-out arithmetic examples, 2,000 standard retrieval examples, and 7,680 semantic retrieval examples, with 2,560 per operator pair. Each semantic pair is immediately followed by the query token; memory lengths 3–6 contribute 640 examples each per pair, with the pair placed at a randomly chosen valid position. Other memory tokens are distinct values from 3–330. The $x\,1\,y$ and $x\,2\,y$ controls each contain 2,560 distinct triples with x, y drawn from 3–330.

Table 12 reproduces the reported hybrid evaluations. One snapshot choice requires explicit accounting: seed 104 is reported at epoch 1,175, selected after observing instability near the end of training. Its epoch-1,200 arithmetic accuracy was 3.60%, following a loss increase from approximately 0.24 at epoch 1,198 to 2.55 and 2.46 in the last two epochs. The earlier evaluation records 99.93% arithmetic accuracy. Consequently, the five-seed summary below uses four epoch-1,200 evaluations and one post-hoc epoch-1,175 evaluation. The earlier seed-104 weights were not retained; its row comes from the saved evaluation log, whereas its saved final predictions belong to epoch 1,200.

Table 11: Model and training hyperparameters of the hybrid toy-model implementation. Batch size counts training sequences per optimizer update; the last batch of each epoch may be smaller. The evaluation epoch selected for each reported run is given in Table 12.

Hyperparameter	Value
Model architecture	
Block count and order	4: linear + convolution, softmax, linear + convolution, softmax
Hidden dimension	256
Attention heads / head dimension	8 / 32
MLP hidden dimension / activation	1,024 / GELU
Vocabulary / maximum sequence length	336 / 12 tokens
Normalization	Pre-LayerNorm in each block; final LayerNorm
Positional encoding	Interleaved RoPE on queries and keys; base 10,000
Convolution type / kernel width	Causal depthwise 1D / 3
Convolution channels / groups	768 / 768 in each linear-attention block; separate query, key, and value channels
Convolution stride / dilation / bias	1 / 1 / enabled; two positions of left zero padding
Linear-attention feature map	$\psi(x) = \mathrm{ELU}(x) + 1$
Attention normalization floor	10^{-6} for the linear-attention denominator
Dropout	0
Embedding / output weights	336×256 embedding; untied output projection
Parameter initialization	PyTorch defaults
Total / convolution parameters	3,338,064 / 6,144
Training	
Random seeds	101, 102, 103, 104, 105
Training examples per seed	270,000 arithmetic + 135,000 retrieval
Training budget	1,200 epochs
Batch size	8,190 sequences per optimizer update
Optimizer	AdamW
AdamW betas / epsilon	$(0.9, 0.999)$ / 10^{-8}
Weight decay / gradient clipping	0.1 / none
Peak learning rate	1×10^{-3}
Warmup	Linear warmup over 500 optimizer updates
Decay schedule / final learning rate	Cosine decay to 0.1 of peak / 1×10^{-4}
Training objective	Weighted causal next-token cross-entropy, normalized by the sum of target weights
Retrieval loss weights	Memory 0.1; query target 2; answer 10
Arithmetic loss weights	Operands 0.1; result 10
Other target weights	Non-PAD targets 1; PAD targets 0

Table 12: Hybrid toy-model results by seed. All performance entries are percentages. `q01`, `q02`, and `q21` give correct predecessor retrieval for the three semantic pairs; First pools first-constituent errors over the three pairs. The dagger marks the post-hoc earlier evaluation described in the text.

Seed	Epoch	Arith.	Standard	`q01`	`q02`	`q21`	First	$x\,1\,y$	$x\,2\,y$
101	1200	100.00	96.60	81.1	67.0	82.7	6.21	100.0	100.0
102	1200	100.00	96.65	83.2	72.5	74.3	15.91	99.7	100.0
103	1200	99.99	99.50	94.1	94.0	95.6	1.73	99.6	100.0
104	1175†	99.93	97.40	50.0	79.6	76.9	19.49	99.8	97.7
105	1200	100.00	98.35	76.9	85.2	81.6	10.01	99.2	86.8

Table 13: Original Transformer reference and aggregate hybrid results. The hybrid summary uses the five evaluations in Table 12, including the earlier seed-104 snapshot. Transformer means come from Tables 3 and 4. Hybrid means are reproduced from the result summary. All values are percentages.

Metric	Transformer	Hybrid
Arithmetic accuracy	99.73	99.99
Standard retrieval accuracy	99.00	97.70
Semantic retrieval accuracy	19.84	79.65
Semantic first-constituent error	46.74	10.67
`q01` accuracy	14.55	77.1
`q01` first-constituent error	38.66	13.1
`q02` accuracy	33.98	79.7
`q02` first-constituent error	36.35	8.1
`q21` accuracy	10.98	82.2
`q21` first-constituent error	65.20	10.8
$x\,1\,y$ accuracy	92.81	99.68
$x\,2\,y$ accuracy	93.82	96.91

Table 13 shows substantially better semantic retrieval and fewer first-constituent errors for the hybrid, with a small reduction in standard retrieval accuracy. The semantic improvement is also visible in each of the four hybrid runs evaluated at epoch 1,200. The Transformer values are the original experiment's reference results; the hybrid uses a reconstructed implementation. The comparison supports the possibility that semantic composition and independently accessible constituent information can coexist in a hybrid model.