What Attention Recalls and Recurrence Controls in Hybrid Language Models

Kirill Afendulev[3,2], **Alexey Dontsov**[3,1], **Elena Tutubalina**[3,1], **Anton Korznikov**[1,3]
[1]HSE University, Moscow, Russia
[2]YSDA, Moscow, Russia
[3]AIRI, Moscow, Russia

Code: https://github.com/kirillTerra/split-prefill

Abstract

Hybrid language models combine attention with a fixed-size recurrent state, but the role of each channel remains unclear. We introduce two cache-level interventions. *Split-prefill* keeps only the KV cache or only the recurrent state from a prefilled context, then generates an answer. *State-swap* pairs the KV cache from one context with the recurrent state from another in a single forward pass. On Qwen3.5 and Falcon-H1, the two channels split sharply by function. Exact retrieval survives only through attention (64–98% of full accuracy) and collapses to zero through recurrence. Output language and persona reverse the pattern: both survive recurrence (70–80% and 3–5×) while KV-only drops to 1% language accuracy. State-swap confirms this causally: the answer takes its value from the KV side and its language from the recurrent side. Recurrent-only generation also accepts words that were never in the context but share meaning or parts with seen items. Attention provides a lookup over what was said; the recurrent state shapes how the model says it next.

1 Introduction

State-space and linear-attention models are a practical alternative to full self-attention for long-context language modelling (Gu and Dao, 2023; Dao and Gu, 2024; Yang et al., 2023; Katharopoulos et al., 2020). Modern open-weight families such as Qwen3.5 (Yang et al., 2025a), Falcon-H1 (Zuo et al., 2025), and Jamba (Lieber et al., 2024) combine softmax attention with a fixed-size recurrent block such as Gated DeltaNet (Yang et al., 2025b) or Mamba (Gu and Dao, 2023). Once such a model is trained, a mechanistic question follows: what does each component actually carry at inference?

To test this we introduce two cache-level interventions (§3). *Split-prefill* keeps either the KV cache or the recurrent state from a context and drops the other; *state-swap* crosses the two caches from different contexts in a single forward pass. We instantiate on Qwen3.5-4B (Gated DeltaNet) and Falcon-H1-3B-Instruct (Mamba).

A natural first guess about the two channels is temporal: attention is precise but expensive long-term memory, the recurrent state is a cheap short-term buffer. This framing turns out to be misleading. Drop the recurrent state and short-range processing still works, but the response policy is gone: language, persona, instruction tone. Dropping the attention KV cache has the opposite effect: response policy is preserved, but the model can no longer report a value for a specific key. The dissociation is functional: it concerns not the age of information each channel retains, but the type of access each channel provides at inference. The recurrent state collapses to zero on exact retrieval, even though stand-alone state-space and linear-attention models retain at least partial recall on the same task families (Arora et al., 2024; Jelassi et al., 2024; Park et al., 2024): in a trained hybrid, the two components separate behaviourally under our interventions.

2 Related Work

Hybrid recurrent–attention LMs. State-space (Gu and Dao, 2023; Dao and Gu, 2024), linear-attention (Katharopoulos et al., 2020; Schlag et al., 2021), RWKV (Peng et al., 2023), GLA (Yang et al., 2023), and Gated DeltaNet (Yang et al., 2025b) models all compress past context into a fixed-size state. Strong recent open-weight models combine such a block with standard attention in a hybrid stack (Lieber et al., 2024; Yang et al., 2025a; Zuo et al., 2025): Jamba interleaves the two block types, Qwen3.5 uses a 3:1 Gated DeltaNet to Gated Attention pattern, and Falcon-H1 adopts a parallel-hybrid block that runs attention and SSMs alongside each other. Reported gains over pure variant follow Waleffe et al. (2024).

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 43888–43900
October 24-29, 2026 ©2026 Association for Computational Linguistics

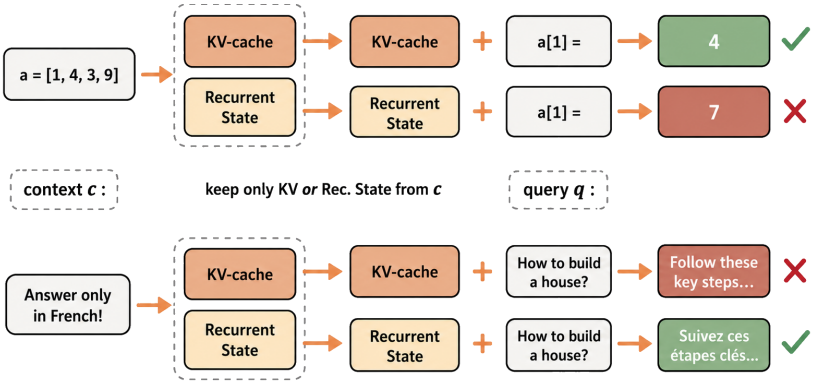

Figure 1: *Split-prefill*. A prompt is split into a context (PROMPT 1) and a query (PROMPT 2). We prefill the context, save its two internal channels (attention KV cache, recurrent state), then continue on the query while keeping only one channel from the context and resetting the other.

Recall as a known weakness of compressed state. Fixed-state models struggle with exact copying, key–value retrieval, and needle-in-a-haystack lookup compared to attention (Arora et al., 2024; Jelassi et al., 2024; Park et al., 2024; Ben-Kish et al., 2025; Liu et al., 2024; Hsieh et al., 2024; Kamradt, 2023). Our results agree on this direction; we make the stronger claim that, the recurrent state does not function as an addressable memory for retrieval at inference. What it does carry is examined in §5.3 and §5.4.

Mechanistic analysis. Where information lives in transformers has been mapped via circuits and activation patching (Elhage et al., 2021; Olsson et al., 2022; Wang et al., 2022; Meng et al., 2022; Geva et al., 2023); Sharma et al. (2024) extend this to Mamba. We complement this line of work by causally swapping the two components of a hybrid on the same prompt.

3 Method

Split-prefill. A prompt is the concatenation of a context c and a query q. A forward pass over c populates the attention KV cache $\mathcal{K}(c)$ and the recurrent state $\mathcal{R}(c)$ (Gated DeltaNet for Qwen, Mamba for Falcon). We then process q in three conditions: FULL (keep both), KV-ONLY (keep $\mathcal{K}$, drop $\mathcal{R}$), and REC-ONLY (keep $\mathcal{R}$, drop $\mathcal{K}$). Because q is always processed normally, the intervention isolates the carry-over from c in a specific channel.

State-swap. We prefill two contexts c_A, c_B of the same format but with different content, then continue a shared query q on the hybrid cache $(\mathcal{K}(c_A), \mathcal{R}(c_B))$ and its mirror. Properties of the answer that follow c_A are attributed to attention; those that follow c_B, to recurrence (Fig. 7, §5.1). Implementation details are in Appendix A.

4 Quantitative Dissociation

Why small, isolated tasks. Standard long-context benchmarks demand token retrieval and response-mode maintenance at once, so component ablations on them (Borobia et al., 2026) cannot attribute any accuracy drop to a specific function. We instead use four small synthetic tasks, each isolating one capability; the trade-off is offset by the size of the effects we report, which are large enough to support a causal finding.

We run split-prefill on Qwen3.5-4B (32 layers, GDN) and Falcon-H1-3B-Instruct (32 layers, Mamba) with four diagnostic tasks, each isolating one capability: **KV retrieve** (a small dictionary; q asks for a value under a key); **list indexing** (a list; q asks for the element at a position); **language following** (an "answer only in L" instruction; q is in a different language; output language scored by a language-ID classifier); and **persona** (c describes a persona; q is neutral; style match scored by an independent LLM judge (Zheng et al., 2023)). We report means over three seeds with 95% hierarchical bootstrap CIs. Dataset sizes, prompts, full CIs across all four models, and the layer-wise sweep modes are in Appendices C and D.

The four tasks separate into two regimes whose ordering of KV-ONLY and REC-ONLY is opposite (Table 1, Fig. 2).

Attention recalls items from prior context. On KV retrieve and list indexing, KV-ONLY retains the bulk of the full-model accuracy (64–98% of baseline across both architectures) while REC-ONLY collapses to zero. The collapse persists in minimal versions of the task (short dictionaries, short lists), ruling out a state-too-small explanation: under split-

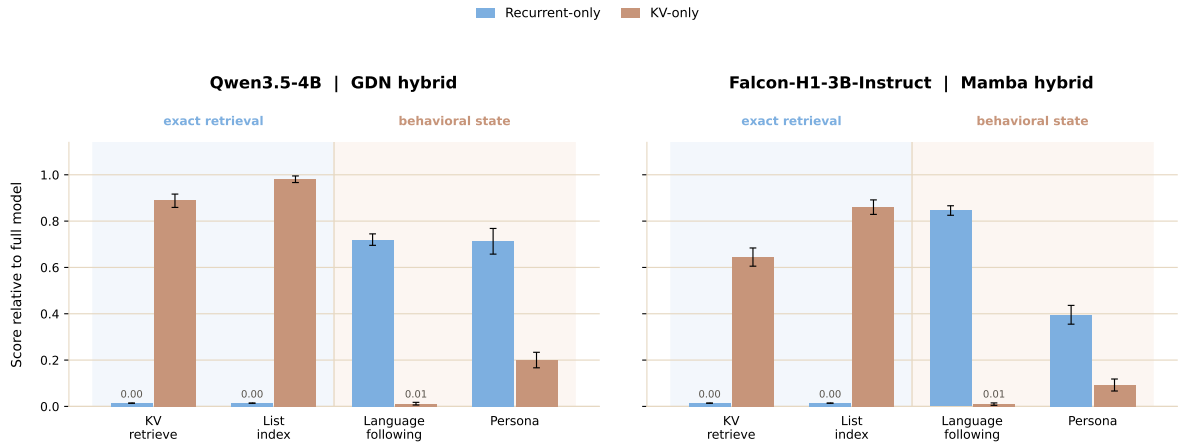

Figure 2: **Main result.** Split-prefill scores on four diagnostic tasks for Qwen3.5-4B (Gated DeltaNet hybrid, left) and Falcon-H1-3B-Instruct (Mamba hybrid, right), normalised to the full-model baseline. Token and position retrieval survive in the attention KV cache and collapse with recurrent state only. Language and persona reverse the pattern: they survive only the recurrent state and are nearly absent from KV alone. Numerical scores are in Table 1.

	Qwen3.5-4B			Falcon-H1-3B		
Task	FULL	REC	KV	FULL	REC	KV
KV retrieve	1.00	0.00	**0.89**	1.00	0.00	**0.64**
List indexing	1.00	0.00	**0.98**	0.96	0.00	**0.83**
Lang. follow	0.97	**0.70**	0.01	0.94	**0.79**	0.01
Persona	0.69	**0.49**	0.14	0.71	**0.28**	0.06

Table 1: Accuracy of the model under split-prefill on four diagnostic tasks. Top two rows: exact retrieval survives only the KV cache. Bottom two: language and persona transfer survive only the recurrent state. REC/KV = REC-ONLY/KV-ONLY. Details in Appendix D.

prefill, recall of specific items from context routes through the attention KV cache, while the recurrent channel produces no retrieval.

Recurrence controls the response mode. On behavioural tasks the ordering flips. REC-ONLY retains 70–80% of the full-model language-following accuracy, while KV-ONLY collapses to ∼1%, a 60–80× gap between the two channels. Persona shows the same direction (REC-ONLY 3–5× higher than KV-ONLY). The KV cache contains the language instruction, yet the model does not act on it: the context establishes a target *mode of continuation* and only the recurrent state controls it.

5 Four Causal Case Studies

We now visualise the dissociation through four causal case studies, ordered by the strength of their argument. The examples shown here are representative; full sample sizes, confidence intervals, and controls are in the appendix referenced at the end of each case study.

5.1 Case Study #1: Mixed Memories

We construct two contexts c_A and c_B that differ on two independent properties: each carries its own small dictionary (so the value stored under a given key is different in the two contexts) and its own *"answer only in L"* instruction (so the requested language is also different). We prefill both contexts, then assemble one mixed cache by taking the KV from c_A together with the recurrent state from c_B, and run a single shared query through this cache in one forward pass (Figure 7, Appendix E).

CTX c_A	dict {MOON → A22, . . . };*answer in French*
CTX c_B	dict {MOON → B33, . . . };*answer in German*
CACHE	$\mathcal{K}(c_A) + \mathcal{R}(c_B)$
QUERY	*What is stored under the key* MOON*?*
ANSWERS	*Der gespeicherte Wert ist* A22.

Figure 3: **Facts from KV, Style from Recurrence.** Hybrid cache $\mathcal{K}(c_A) + \mathcal{R}(c_B)$ yields the value from c_A and the language from c_B.

The model returns one fluent sentence in which the retrieved value is the one stored in the KV-source context c_A, and the language is the one requested by the recurrent-source context c_B. The two properties are jointly correct on $\geq$ 99% of items in both models; same-source controls ($\mathcal{K}_A + \mathcal{R}_A$, $\mathcal{K}_B + \mathcal{R}_B$) match FULL. Per-direction breakdown and CIs: Appendix E.

5.2 Case Study #2: Recall, but Not Control

A starker test: ask the model under KV-ONLY to *name* the requested language. The KV cache con-

CONTEXT *Answer only in French.*

QUERY *Hello, how are you?*

ANSWERS

KV-ONLY → *Hello!* REC-ONLY → *Bonjour !*

QUERY *What is your language for answer?*

ANSWERS

KV-ONLY → *French.* REC-ONLY → *Français.*

Figure 4: **Recall ≠ Control.** KV-only names the requested language correctly but answers in English; rec-only reverses the pattern.

tains the relevant tokens, so the model can retrieve them; but retrieving the name and acting on the instruction are different things.

We pair every language setting with both a *label* query (what is the requested language?) and a *behavioural* query (a normal English question). Under KV-ONLY, models name the language correctly but answer in English; under REC-ONLY, the ordering inverts. The KV cache lets the model report what the instruction was. To actually follow it, the recurrent state has to be carried over from the context.

5.3 Case Study #3: Associative False Recall

If the recurrent state is not an addressable memory, what does it preserve? We use a DRM-style (Roediger and McDermott, 1995) list of words strongly associated with an absent target.

CONTEXT

bed, rest, pillow, dream, blanket, tired, night, snore, nap

QUERY

Was the exact word "sleep" in the list? Answer yes or no. (gold: *no*)

ANSWERS

KV-ONLY → *No.* REC-ONLY → *Yes.*

Figure 5: **False Familiars.** Rec-only falsely accepts an associated but absent target; KV-only correctly rejects.

Under KV-ONLY, both models reject the trap; under REC-ONLY, false-yes rates rise sharply on the associated list while a matched *neutral-absent* control stays near floor. The recurrent state preserves the *semantic field* of seen words rather than their identities: a target that was never tokenised in context is nevertheless "familiar" enough to be reported as present. Controls and per-seed rates: Appendix G.

CONTEXT

sunflower, moonlight, blueberry, footprint, . . .

QUERY

Was the exact word "sunlight" in the list? (gold: *no*)

ANSWERS

KV-ONLY → *No.* REC-ONLY → *Yes.*

Figure 6: **Broken Binding.** "sun-" (from sunflower) and "-light" (from moonlight) both appear in the list; the compound "sunlight" does not.

5.4 Case Study #4: Conjunction Trap

A complementary failure mode is *memory-conjunction errors* (Reinitz et al., 1992): features of two seen items recombine into an unseen item that is mistaken for an original. If the recurrent state stores feature-level rather than item-level evidence, it should fall for this trap.

Under KV-ONLY, both models again reject the unseen recombination. Under REC-ONLY, false-yes rates rise sharply on conjunction items but stay near floor on a *single-part-absent* control where only one of the two morphemes is present: the recurrence signal is driven by the *simultaneous* presence of both parts. Component features of stored items survive in the recurrent state; the binding that ties them together as a specific item is what gets lost. Full controls: Appendix H.

6 Conclusion

In a trained hybrid LM, the two components do not differ in how long they retain past context. They differ in how the model accesses what they hold. The KV cache works as an addressable store: the model can pull specific items out of it. The recurrent state works as a compressed prior that shapes language, persona, and semantic field of generation. The claim is functional. It concerns what each channel causally contributes to generation, not what each channel statically encodes. State-swap makes the attribution causal at the level of a single answer. The pattern parallels the explicit/implicit memory distinction in cognitive psychology (Tulving, 1985; Schacter, 1987) and refines the recall-deficiency story for state-space and linear-attention models (Arora et al., 2024; Jelassi et al., 2024; Park et al., 2024): the recurrent state should be read as a different kind of memory, and benchmarks that score addressable lookup and behavioural conditioning together cannot show this.

Limitations

Functional, not static, content. Our results characterise what each channel *causally drives* at inference, not what it *statically encodes*. We do not claim either channel is empty of the other type of content. Demonstration 5.2 is direct evidence to the contrary: the KV cache lets the model name the requested language on demand, even when under KV-ONLY it cannot make the model answer in that language.

Probing left for future work. Mapping the static information content of attention and recurrent representations would require trained probing classifiers on activations from each channel. We leave this analysis to future work.

Nature of the intervention. Split-prefill resets one cache while preserving the other, which produces a partly out-of-distribution combination: the two channels are normally co-updated at every position, and resetting one breaks a coherence the model relies on. Part of the rec-only / kv-only collapse could be due to this incoherence, not to the channel itself being uninformative. State-swap (Demo 5.1) addresses this concern: both caches come from real prefills of well-formed contexts; only the source contexts differ, so neither channel is in a degenerate state. The same dissociation holds in this closer-to-natural setting, which is harder to attribute to the intervention itself.

Acknowledgements

The study was implemented in the framework of the Basic Research Program at HSE University (HSE-BR-2025-025). We also acknowledge the computational resources of the HPC facilities at HSE University.

References

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Y Zou, Atri Rudra, and Christopher Ré. 2024. Zoology: Measuring and improving recall in efficient language models. In *International conference on learning representations*, volume 2024, pages 15664–15730.

Assaf Ben-Kish, Itamar Zimerman, Shady Abu-Hussein, Nadav Cohen, Amir Globerson, Lior Wolf, and Raja Giryes. 2025. Decimamba: Exploring the length extrapolation potential of mamba. In *International Conference on Learning Representations*, volume 2025, pages 101148–101170.

Hector Borobia, Elies Seguí-Mas, and Guillermina Tormo-Carbó. 2026. Functional component ablation reveals specialization patterns in hybrid language model architectures. *arXiv preprint arXiv:2603.22473*.

Tri Dao and Albert Gu. 2024. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, and 1 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Samy Jelassi, David Brandfonbrener, Sham M Kakade, and Eran Malach. 2024. Repeat after me: Transformers are better than state space models at copying. *arXiv preprint arXiv:2402.01032*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomáš Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th conference of the European chapter of the association for computational linguistics: volume 2, short papers*, pages 427–431.

Greg Kamradt. 2023. Needle in a haystack – pressure testing LLMs.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, and 1 others. 2024. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, and 1 others. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

Jongho Park, Jaeseung Park, Zheyang Xiong, Nayoung Lee, Jaewoong Cho, Samet Oymak, Kangwook Lee, and Dimitris Papailiopoulos. 2024. Can mamba learn how to learn? a comparative study on in-context learning tasks. *arXiv preprint arXiv:2402.04248*.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, and 1 others. 2023. Rwkv: Reinventing rnns for the transformer era. In *Findings of the association for computational linguistics: EMNLP 2023*, pages 14048–14077.

Mark Tippens Reinitz, William J Lammers, and aBarbara Pitt's Cochran. 1992. Memory-conjunction errors: Miscombination of stored stimulus features can produce illusions of memory. *Memory & Cognition*, 20(1):1–11.

Henry L Roediger and Kathleen B McDermott. 1995. Creating false memories: Remembering words not presented in lists. *Journal of experimental psychology: Learning, Memory, and Cognition*, 21(4):803.

Daniel L Schacter. 1987. Implicit memory: History and current status. *Journal of experimental psychology: learning, memory, and cognition*, 13(3):501.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. 2021. Linear transformers are secretly fast weight programmers. In *International conference on machine learning*, pages 9355–9366. PMLR.

Arnab Sen Sharma, David Atkinson, and David Bau. 2024. Locating and editing factual associations in mamba. *arXiv preprint arXiv:2404.03646*.

Endel Tulving. 1985. How many memory systems are there? *American psychologist*, 40(4):385.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, and 1 others. 2024. An empirical study of mamba-based language models. *arXiv preprint arXiv:2406.07887*.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. 2025b. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations*, volume 2025, pages 29687–29707.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. 2023. Gated linear attention transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Jingwei Zuo, Maksim Velikanov, Ilyas Chahed, Younes Belkada, Dhia Eddine Rhayem, Guillaume Kunsch, Hakim Hacid, Hamza Yous, Brahim Farhat, Ibrahim Khadraoui, and 1 others. 2025. Falcon-h1: A family of hybrid-head language models redefining efficiency and performance. *arXiv preprint arXiv:2507.22448*.

A Implementation Details

Cache extraction. We extract caches at the end of the context prefill before any query token is processed. For Qwen3.5-4B we use the HuggingFace `transformers` integration of Gated DeltaNet; for Falcon-H1 we use the official HuggingFace release.

B Split-prefill: Detailed Procedure

Notation. A hybrid model has L layers partitioned into an attention set $\mathbb{A}$ and a recurrent set $\mathbb{R}$. Running the model over a token sequence c populates a memory state that factorises across the two sets:

$$M(c) = \big(\underbrace{\{(K_\ell, V_\ell)\}_{\ell\in\mathbb{A}}}_{\mathcal{K}(c)}, \underbrace{\{(S_\ell, C_\ell)\}_{\ell\in\mathbb{R}}}_{\mathcal{R}(c)}\big),$$

where $K_\ell, V_\ell \in \mathbb{R}^{h\times|c|\times d}$; S_ℓ is the layer's recurrent state (an SSM state for Mamba-style layers, a matrix-valued state for delta-rule and gated-linear-attention layers); and C_ℓ is the short causal convolution's state where the architecture has one. C_ℓ is treated as part of the recurrent channel throughout: carried with S_ℓ, dropped with it. The KV channel grows with $|c|$ and is content-addressable; the recurrent channel is a fixed-size summary. This asymmetry is what split-prefill probes.

The three conditions. Every example is a pair (c, q): a context carrying the information under test and a query identical across conditions. We run the model over c alone, retain $M(c)$, construct a fresh memory state, populate one or both channels, and then run over q:

- FULL: $M \leftarrow (\mathcal{K}(c),\ \mathcal{R}(c))$; no intervention.
- KV-ONLY: $M \leftarrow (\mathcal{K}(c),\ \emptyset)$; the recurrent state is zeroed.
- REC-ONLY: $M \leftarrow (\emptyset,\ \mathcal{R}(c))$; the KV cache is dropped.

Three properties matter. First, the context is prefilled once; the ablated conditions reuse the state from the same forward pass over c and do not rerun the model on a truncated context. Second, the query is byte-identical across conditions and tokenised once. Third, positions follow the retained KV: under FULL and KV-ONLY, q begins at position $|c|$; under REC-ONLY the KV is empty so q begins at 0. This is deliberate. A dropped cache has no positions to occupy, and padding with zero keys to equalise positions would introduce a third, uncontrolled condition, since zero keys are still attended to.

Constructing c and q. The split must land inside a single user turn, so that no template markup falls between c and q and the boundary is not a role change. We render the whole conversation once, $T = \text{template}(\{\text{user} : u_1 \| u_2\},\ \text{add_generation_prompt})$, locate the first occurrence of u_1 in T, and cut immediately after it. Cutting a rendered string rather than concatenating two separately rendered fragments guarantees that $c \| q$ is exactly what the model would see without intervention, so the FULL condition coincides with ordinary inference. For reasoning models we append an empty `<think></think>` block to the assistant opener; without it every condition answers in English reasoning prose, which destroys the language metric.

The recurrent-state carry hazard. Most public implementations of hybrid layers only consume a restored recurrent state on the single-token decode path. A common gate reads

```
use_precomputed_states = cache is not None
    and cache.has_previous_state(l)
    and seq_len == 1
```

so when q is fed as one multi-token chunk the restored S_ℓ is silently ignored and the scan restarts from zero. No error is raised. REC-ONLY then measures a model with no context at all, and its collapse to chance is easily misread as evidence that the recurrent channel carries nothing. Two equivalent remedies, chosen per model: patch the mixer to pass the restored state as an `initial_state` argument to the chunked kernel, or feed q token-by-token so the stock gate admits it.

Verification protocol (mandatory per model). Because the failure above is silent, we verify before recording any number that the restored state actually influences the output. Holding everything else fixed:

$$\Delta \;=\; \left\| \text{logits}(q \mid \mathcal{R}(c)) \;-\; \text{logits}(q \mid \mathbf{0}) \right\|_\infty .$$

$\Delta \approx 0$ proves the state never reached the model and invalidates every REC-ONLY and crossed-cache number for that configuration; the run is discarded rather than reported. We observed exactly this on one model under chunked prefill ($\Delta = 0.0000$), fixed by the token-wise remedy ($\Delta = 12.1$). Two cautions. First, healthy controls do not substitute for this check: in the crossed-cache experiment the same-source controls can look perfectly healthy while the state is dropped, because with both channels from a single context the KV alone explains the outputs. Second, verification must run inside the exact code path used for the reported numbers, including the specific model class and library version.

Crossed cache (state-swap). The causal variant pairs contexts c_A, c_B differing in both retrieved content and instructed response mode, assembles $M \leftarrow (\mathcal{K}(c_A),\ \mathcal{R}(c_B))$, and runs the shared query. Same-source assemblies serve as controls and bound achievable accuracy. We assert at run time that q renders identically for both. For implementations taking an explicit starting position, the position passed with q is the length of the KV-source context, keeping positions consistent with the restored keys.

Decoding and scoring. Greedy (argmax), no sampling, stopping at EOS or after 8 new tokens for retrieval (single-word answers) and 48 for free-text. Retrieval is scored by exact match of the first alphabetic word. Language is scored with an off-the-shelf identifier restricted to the target set, so a confident prediction outside it counts as failure rather than being silently mapped. Persona is scored by an instruction-tuned LLM judge emitting a binary verdict, held fixed across conditions and

models. Since all conditions share query, prompts, and decoder, differences cannot arise from decoding.

Per-model notes. A per-model summary records where the state is stored, which carry remedy was used, and the verified Δ. Two practical hazards. Several checkpoints ship custom modelling code pinned to an older library version, and running them under a newer one can fail silently, producing fluent-looking but degenerate output in every condition rather than an exception. Each such model is therefore run under the library version its code targets, with raw generations inspected before scoring. A renamed dtype or position argument across versions fails silently as well. Both hazards are caught by the verification above combined with a visual check of the control generations.

C Datasets

All datasets are programmatically generated.

KV retrieve. A dictionary with 15–30 entries; keys and values are short ASCII tokens unlikely to overlap with each other. The query asks for the value under a single key. Gold: the exact value as a string. The lower bound on entry count ensures retrieval is non-trivial; the upper bound keeps total length manageable.

List indexing. A Python-style list of 4–10 short tokens. The query asks for the element at a numeric position. Gold: the exact list element.

Language following. 18 target languages (EN, FR, DE, ES, IT, PT, NL, RU, PL, TR, UK, SV, DA, JA, ZH, KO, HI, ID). The context contains the instruction "Answer only in L"; the query is a meaning-preserving English question. We score with `fastText` `lid.176` (Joulin et al., 2017), taking the highest-probability language label of the generation.

Persona. 15 persona descriptions (pirate, cosmonaut, wizard, samurai, knight, vampire, alien, surfer, caveman, sage, robot, detective, ninja, shaman, cowboy) paired with 15 neutral questions. An independent judge LM (Qwen2.5-72B-Instruct, specifically chosen to be in a different family than the tested models) returns a $[0, 1]$ persona-match score.

Association trap. 100 DRM-style associated lists of 8–10 words each, drawn from the Roediger–McDermott norms; plus 100 matched neutral controls (unrelated words) and 100 positive-present controls (target word actually placed in the list).

Conjunction trap. 200 conjunction items (e.g. *sunflower* + *moonlight* $\rightarrow$ probe *sunlight*). Three matched controls: *neutral-absent* (target unrelated to any list item), *single-part-absent* (only one component morpheme present in the list), and *positive-present* (the conjunction item itself in the list).

Crossed dictionary + language (Demo 1). 100 crossed examples per direction ($\mathcal{K}_A + \mathcal{R}_B$ and the mirror) plus 200 matched same-source controls ($\mathcal{K}_A + \mathcal{R}_A$ and $\mathcal{K}_B + \mathcal{R}_B$). L_A, L_B drawn from {fr, de, es, it, ja, ko}.

D Quantitative Dissociation: Full Results

Table 2 reproduces Table 1 across all four evaluated models with 95% hierarchical bootstrap CIs (3 seeds, 10,000 replicates, resampling seeds then examples within seed). The two recurrent-only retrieval cells are 0/600 on every seed for every model. The two recurrent-only behavioural cells substantially exceed kv-only on every model with the single exception of Falcon-H1-7B persona (kv-only 0.29, rec-only 0.25; paired bootstrap -3.8 p.p. $[-6.9, -0.6]$, marked †). Across both families, kv-only retrieval accuracy improves monotonically with model size while the recurrent-only collapse persists, preserving the dissociation.

Table 3 reports Qwen-specific layer-wise sweep modes: LOWER-KV+FULL-REC keeps the attention KV cache only for the bottom half of attention layers and the recurrent state in full; FULL-KV+LOWER-REC keeps the KV cache in full and the recurrent state only for the lower half of GDN layers. Retrieval is preserved only when KV is kept in full; behavioural transfer is preserved whenever the recurrent state is kept in full. These modes were not implemented for Falcon-H1.

Persona judge. Qwen2.5-72B-Instruct with three samples per response. We use no model from the tested families (Qwen3.5 or Falcon-H1) to avoid self-judging.

E Demonstration 1 — State-swap

Construction. 100 crossed examples per direction ($\mathcal{K}_A + \mathcal{R}_B$, $\mathcal{K}_B + \mathcal{R}_A$; n=200 total crossed per seed) plus 200 matched same-source controls ($\mathcal{K}_A + \mathcal{R}_A$, $\mathcal{K}_B + \mathcal{R}_B$). Match criterion for *value→KV*: case-insensitive substring match of the

Model	Task	FULL	REC-ONLY	KV-ONLY
Qwen3.5-4B	KV retrieve	1.00 [0.99, 1.00]	0.00 [0.00, 0.00]	0.89 [0.86, 0.92]
	List index	1.00 [0.99, 1.00]	0.00 [0.00, 0.00]	0.98 [0.96, 0.99]
	Lang. foll.	0.97 [0.96, 0.98]	0.70 [0.67, 0.72]	0.01 [0.01, 0.02]
	Persona	0.69 [0.66, 0.72]	0.49 [0.46, 0.53]	0.14 [0.12, 0.16]
Qwen3.5-9B	KV retrieve	1.00 [0.99, 1.00]	0.00 [0.00, 0.00]	0.94 [0.92, 0.96]
	List index	1.00 [0.99, 1.00]	0.00 [0.00, 0.00]	0.99 [0.98, 1.00]
	Lang. foll.	0.96 [0.94, 0.97]	0.62 [0.60, 0.65]	0.01 [0.00, 0.01]
	Persona	0.74 [0.72, 0.77]	0.63 [0.60, 0.66]	0.08 [0.07, 0.10]
Falcon-H1-3B	KV retrieve	1.00 [0.99, 1.00]	0.00 [0.00, 0.00]	0.64 [0.60, 0.68]
	List index	0.96 [0.94, 0.97]	0.00 [0.00, 0.00]	0.83 [0.79, 0.86]
	Lang. foll.	0.94 [0.93, 0.95]	0.79 [0.77, 0.82]	0.01 [0.00, 0.01]
	Persona	0.71 [0.68, 0.73]	0.28 [0.25, 0.31]	0.06 [0.05, 0.08]
Falcon-H1-7B	KV retrieve	1.00 [1.00, 1.00]	0.00 [0.00, 0.00]	0.99 [0.98, 1.00]
	List index	1.00 [1.00, 1.00]	0.00 [0.00, 0.00]	0.98 [0.96, 0.99]
	Lang. foll.	0.92 [0.91, 0.93]	0.61 [0.59, 0.64]	0.05 [0.04, 0.06]
	Persona	0.75 [0.72, 0.78]	0.25 [0.22, 0.28]	0.29 [0.26, 0.32]†
Qwen3.5-27B‡	KV retrieve	1.00 [1.00, 1.00]	0.00 [0.00, 0.00]	0.99 [0.98, 1.00]
	List index	1.00 [1.00, 1.00]	0.00 [0.00, 0.00]	1.00 [1.00, 1.00]
	Lang. foll.	0.99 [0.98, 1.00]	0.83 [0.79, 0.86]	0.08 [0.06, 0.11]
	Persona	0.74 [0.69, 0.78]	0.66 [0.61, 0.71]	0.31 [0.26, 0.36]
Falcon-H1-34B‡	KV retrieve	1.00 [1.00, 1.00]	0.00 [0.00, 0.00]	0.85 [0.80, 0.90]
	List index	1.00 [1.00, 1.00]	0.02 [0.00, 0.04]	0.96 [0.93, 0.99]
	Lang. foll.	0.96 [0.94, 0.98]	0.41 [0.36, 0.46]	0.25 [0.21, 0.29]
	Persona	0.80 [0.75, 0.84]	0.18 [0.14, 0.23]	0.33 [0.28, 0.39]†
Kimi-Linear-48B*	KV retrieve	0.99	0.00	0.94
	List index	0.95	0.00	0.91
	Lang. foll.	0.94	0.77	0.00
	Persona	0.99	0.80	0.14
Olmo-Hybrid-7B*	KV retrieve	0.99	0.00	0.99
	List index	0.98	0.00	0.91
	Lang. foll.	0.97	0.72	0.01
	Persona	0.99	0.94	0.08
Jamba2-3B*	KV retrieve	0.59	0.00	0.77
	List index	0.79	0.00	0.40
	Lang. foll.	0.63	0.55	0.07
	Persona	0.98	0.75	0.36

Table 2: Main split-prefill results across all evaluated models. Rows without markers: means over 3 seeds with 95% hierarchical bootstrap CIs in brackets. n=200 per seed for retrieval, 450 for language, 225 for persona. †Cells where KV-ONLY exceeds REC-ONLY on a behavioural task; see text. ‡Single-seed extension (seed 42 only); intervals are within-seed bootstraps and do not quantify across-seed variability. *Additional architectures; point estimates only (bootstrap CIs not computed for these runs).

Task	lower KV + full rec	full KV + lower rec
KV retrieve	0.02 [0.01, 0.03]	1.00 [0.99, 1.00]
List indexing	0.01 [0.00, 0.02]	1.00 [0.99, 1.00]
Lang. follow	0.95 [0.94, 0.96]	0.71 [0.68, 0.73]
Persona	0.68 [0.65, 0.71]	0.65 [0.62, 0.68]

Table 3: Layer-wise sweep on Qwen3.5-4B. The two modes dissociate layer-wise as well as channel-wise: retrieval requires KV in full, behavioural transfer requires recurrent state in full.

KV-source dictionary value in the output. Match criterion for *lang→rec*: `fastText lid.176` identifies the recurrent-source language as the top-probability label.

Results. Table 4 reports both crossing directions for each model. On Qwen-4B and Qwen-9B every crossed example is jointly correct. Falcon-H1-3B is near-perfect ($\geq$ 0.99 both-correct in either direction). Falcon-H1-7B retains perfect value retrieval but loses some language behaviour ($\approx$ 0.87–0.89 both-correct), consistent with its weaker recurrent-only language-following score in Table 2. Same-source controls ($\mathcal{K}_A + \mathcal{R}_A$ and $\mathcal{K}_B + \mathcal{R}_B$) reach 1.00 both-correct for Qwen-4B, Qwen-9B and Falcon-H1-7B; Falcon-H1-3B con-

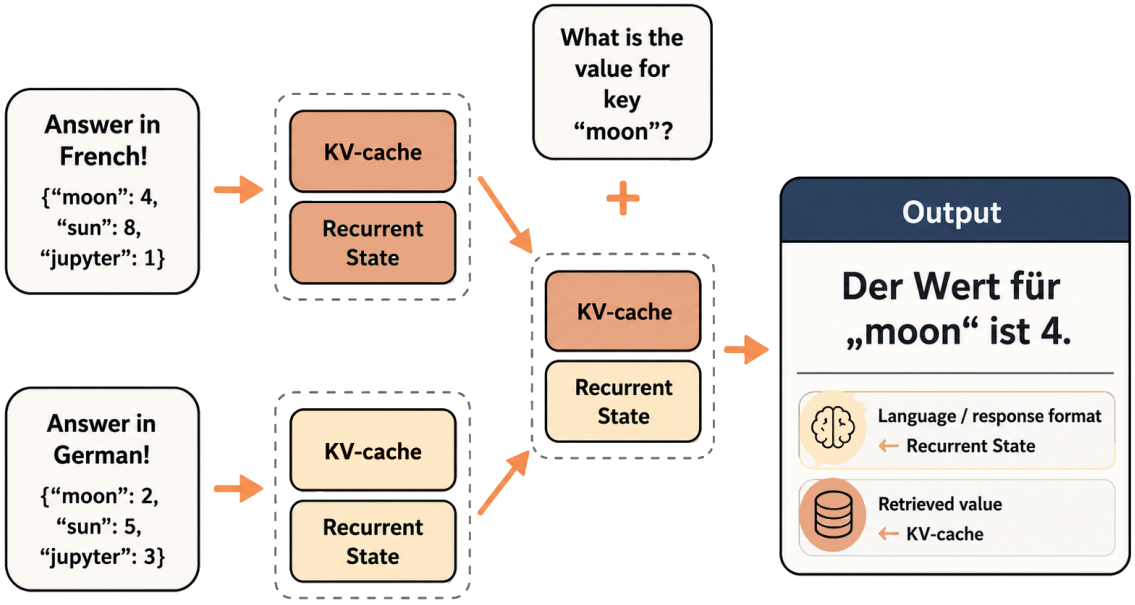

Figure 7: State-swap construction. Two contexts c_A and c_B each pair a dictionary with a language instruction; prefilling each yields its own KV cache and recurrent state. We assemble a hybrid cache (KV from c_A + recurrent state from c_B), feed a shared query, and read off which channel each property of the answer follows. In the illustrated cross, the answer language (German) follows the recurrent source c_B while the dictionary value is retrieved from the KV source c_A.

trols are also near-perfect (~ 0.99).

F Demonstration 2 — Label vs. behavioural language

Construction. A 20-language self-report probe of the form *“Answer only in L. What is your language for answer? Answer in one word”* (n=20 per seed, 3 seeds). *English label*: the output is the English name of L (e.g. *Russian*); this exposes literal label retrieval from the KV cache. *Target-language label*: the output is the language name in L itself (e.g. *Russkij* for Russian or *Deutsch* for German); this requires the model to also enact the language behaviour.

Results. Table 5 shows both scores together. Qwen3.5-9B is the cleanest case: KV-ONLY names the language in English 90% of the time while almost never speaking it (0.05 target-label), and REC-ONLY reverses the pattern (0.10 English / 0.95 target). Qwen3.5-4B and Falcon-H1-7B show the same direction. Falcon-H1-3B is weaker on this specific self-report prompt despite strong ordinary language-following accuracy (0.79 REC-ONLY vs. 0.01 KV-ONLY on the 450-item dataset, Table 1); the broader argument rests on the larger language-following dataset and on the three other models here.

G Demonstration 3 — Associative (DRM) trap

Construction. Per seed: 100 associated lists (Roediger–McDermott norms; 8–10 words each, target word absent), 100 matched neutral lists (target unrelated to list contents, absent), 100 positive-present lists (target word actually placed at a random position). The query in every condition is *“Was the exact word ‘w’ in the list?”*. False-positive rate (FPR) is the fraction of *yes* responses on absent-target conditions; true positive rate (TPR) is the fraction of *yes* on the present condition.

Results. Table 6 shows the three rates per model and mode. REC-ONLY drives a large associated-absent FPR on every model (Qwen-4B 0.87, Qwen-9B 0.89, Falcon-3B 0.45, Falcon-7B 0.99). For three of four models the neutral control stays at or below 0.10, so the false yes is specifically driven by the associative cue rather than a blanket yes-bias. Falcon-H1-7B is the exception: its REC-ONLY neutral FPR is 0.18 and KV-ONLY associated/neutral are both above 0.55, indicating broad

Model	Crossed state	value→KV	lang→rec	both
Qwen3.5-4B	$\mathcal{K}_A + \mathcal{R}_B$	1.00 [1.00, 1.00]	1.00 [1.00, 1.00]	1.00 [1.00, 1.00]
	$\mathcal{K}_B + \mathcal{R}_A$	1.00 [1.00, 1.00]	1.00 [1.00, 1.00]	1.00 [1.00, 1.00]
Qwen3.5-9B	$\mathcal{K}_A + \mathcal{R}_B$	1.00 [1.00, 1.00]	1.00 [1.00, 1.00]	1.00 [1.00, 1.00]
	$\mathcal{K}_B + \mathcal{R}_A$	1.00 [1.00, 1.00]	1.00 [1.00, 1.00]	1.00 [1.00, 1.00]
Falcon-H1-3B	$\mathcal{K}_A + \mathcal{R}_B$	1.00 [0.99, 1.00]	1.00 [0.99, 1.00]	0.99 [0.98, 1.00]
	$\mathcal{K}_B + \mathcal{R}_A$	1.00 [0.99, 1.00]	0.99 [0.98, 1.00]	0.99 [0.98, 1.00]
Falcon-H1-7B	$\mathcal{K}_A + \mathcal{R}_B$	1.00 [1.00, 1.00]	0.89 [0.85, 0.93]	0.89 [0.85, 0.93]
	$\mathcal{K}_B + \mathcal{R}_A$	1.00 [1.00, 1.00]	0.87 [0.83, 0.91]	0.87 [0.83, 0.91]
Qwen3.5-27B‡	$\mathcal{K}_A + \mathcal{R}_B$	1.00 [1.00, 1.00]	0.96 [0.92, 0.99]	0.96 [0.92, 0.99]
	$\mathcal{K}_B + \mathcal{R}_A$	1.00 [1.00, 1.00]	0.95 [0.90, 0.99]	0.95 [0.90, 0.99]
Falcon-H1-34B‡	$\mathcal{K}_A + \mathcal{R}_B$	1.00 [1.00, 1.00]	0.72 [0.63, 0.81]	0.72 [0.63, 0.81]
	$\mathcal{K}_B + \mathcal{R}_A$	1.00 [1.00, 1.00]	0.71 [0.62, 0.80]	0.71 [0.62, 0.80]
Kimi-Linear-48B*	$\mathcal{K}_A + \mathcal{R}_B$	1.00	1.00	1.00
	$\mathcal{K}_B + \mathcal{R}_A$	1.00	0.98	0.98
Olmo-Hybrid-7B*	$\mathcal{K}_A + \mathcal{R}_B$	1.00	0.96	0.96
	$\mathcal{K}_B + \mathcal{R}_A$	1.00	0.94	0.94
Jamba2-3B*	$\mathcal{K}_A + \mathcal{R}_B$	1.00	0.92	0.92
	$\mathcal{K}_B + \mathcal{R}_A$	1.00	0.90	0.90

Table 4: State-swap (Demo 1). Each row is one crossing direction. Unmarked rows: pooled across 3 seeds, n=100 examples per seed per direction (n=300 total per row). ‡Single-seed extension (seed 42 only); intervals are within-seed bootstraps. *Additional architectures; point estimates only. For Jamba2-3B the same-source controls reach only 0.86–0.89 both-correct.

Model	Mode	Eng. label	Target-lang. label
Qwen3.5-4B	FULL	0.15 [0.07, 0.25]	0.90 [0.82, 0.97]
	REC-ONLY	0.05 [0.00, 0.12]	0.95 [0.88, 1.00]
	KV-ONLY	0.50 [0.38, 0.63]	0.05 [0.00, 0.12]
Qwen3.5-9B	FULL	0.05 [0.00, 0.12]	1.00 [1.00, 1.00]
	REC-ONLY	0.10 [0.03, 0.18]	0.95 [0.88, 1.00]
	KV-ONLY	0.90 [0.82, 0.97]	0.05 [0.00, 0.12]
Falcon-H1-3B	FULL	0.05 [0.00, 0.12]	0.85 [0.75, 0.93]
	REC-ONLY	0.00 [0.00, 0.00]	0.10 [0.03, 0.18]
	KV-ONLY	0.55 [0.43, 0.67]	0.40 [0.28, 0.52]
Falcon-H1-7B	FULL	0.10 [0.03, 0.18]	0.95 [0.88, 1.00]
	REC-ONLY	0.05 [0.00, 0.12]	0.50 [0.37, 0.62]
	KV-ONLY	1.00 [1.00, 1.00]	0.05 [0.00, 0.12]
Qwen3.5-27B‡	FULL	0.05	1.00
	REC-ONLY	0.05	0.95
	KV-ONLY	0.85	0.20
Falcon-H1-34B‡	FULL	0.15	0.85
	REC-ONLY	0.65	0.10
	KV-ONLY	0.35	0.05

Table 5: Self-report probe (Demo 2). Fraction of responses that contain the English language name (col. 3) or the language name in the target language (col. 4). Unmarked rows: n=20 per seed, 3 seeds, 95% bootstrap CIs. ‡Single-seed extension (seed 42 only, one example per target language); point estimates shown without intervals.

false-membership susceptibility rather than a clean associative effect; the rec-only associated rate still exceeds the rec-only neutral rate by 0.81 p.p., but the interpretation should be more guarded for this model.

H Demonstration 4 — Memory-conjunction trap

Construction. Per seed: 200 conjunction items, each a list of 8–12 single-token words whose subset includes both morphemes of a target compound

Model	Mode	assoc-absent FPR	neutral-absent FPR	present TPR
Qwen3.5-4B	FULL	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	0.97 [0.94, 0.99]
	REC-ONLY	0.87 [0.82, 0.91]	0.10 [0.07, 0.14]	0.98 [0.96, 1.00]
	KV-ONLY	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	0.71 [0.65, 0.77]
Qwen3.5-9B	FULL	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	1.00 [1.00, 1.00]
	REC-ONLY	0.89 [0.84, 0.93]	0.01 [0.00, 0.02]	0.93 [0.90, 0.96]
	KV-ONLY	0.01 [0.00, 0.03]	0.00 [0.00, 0.00]	0.86 [0.81, 0.92]
Falcon-H1-3B	FULL	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	1.00 [0.99, 1.00]
	REC-ONLY	0.45 [0.38, 0.52]	0.00 [0.00, 0.01]	0.72 [0.66, 0.77]
	KV-ONLY	0.05 [0.02, 0.09]	0.01 [0.00, 0.02]	0.71 [0.64, 0.77]
Falcon-H1-7B	FULL	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	1.00 [1.00, 1.00]
	REC-ONLY	0.99 [0.97, 1.00]	0.18 [0.14, 0.23]	0.99 [0.98, 1.00]
	KV-ONLY	0.60 [0.55, 0.66]	0.56 [0.49, 0.62]	1.00 [0.99, 1.00]
Qwen3.5-27B‡	FULL	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	1.00 [1.00, 1.00]
	REC-ONLY	0.29 [0.20, 0.38]	0.01 [0.00, 0.03]	0.48 [0.38, 0.58]
	KV-ONLY	0.29 [0.20, 0.38]	0.20 [0.13, 0.28]	1.00 [1.00, 1.00]
Falcon-H1-34B‡	FULL	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	1.00 [1.00, 1.00]
	REC-ONLY	0.72 [0.63, 0.81]	0.00 [0.00, 0.00]	1.00 [1.00, 1.00]
	KV-ONLY	0.32 [0.23, 0.41]	0.11 [0.05, 0.18]	0.86 [0.79, 0.92]

Table 6: DRM trap (Demo 3). Three rates per (model, mode): associated-absent false-positive (the trap), neutral-absent false-positive (yes-bias control), and present true-positive (verifies the model can affirm true membership). Unmarked rows: n=100 per condition per seed, 3 seeds, 95% bootstrap CIs. ‡Single-seed extension (seed 42 only); intervals are within-seed bootstraps.

(e.g. *sunflower* and *moonlight* for the probe *sunlight*). Four matched controls: *neutral-absent* (target unrelated to any list item), *left-only* and *right-only* (only one of the two morphemes present in the list), and *positive-present* (the conjunction itself in the list).

Results. Table 7 reports all five rates per (model, mode). On all four models the REC-ONLY conjunction-absent rate exceeds both single-part-absent rates: Qwen-4B 0.32 vs. $0.18/0.05$; Qwen-9B 0.17 vs. $0.08/0.03$; Falcon-3B 0.15 vs. $0.06/0.04$; Falcon-7B 0.75 vs. $0.61/0.48$. The signal therefore amplifies when both component morphemes are simultaneously present, beyond what either part alone produces. Falcon-H1-7B again shows broad recurrent-only false-positive susceptibility (neutral 0.68, both single-parts above 0.48); the conjunction effect persists relatively, but absolute rates are inflated.

Model	**Mode**	conj. FPR	neutral FPR	left-only FPR	right-only FPR	present TPR
Qwen3.5-4B	FULL	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	1.00 [1.00, 1.00]
	REC-ONLY	0.32 [0.28, 0.36]	0.10 [0.07, 0.12]	0.18 [0.15, 0.22]	0.05 [0.04, 0.07]	0.53 [0.49, 0.57]
	KV-ONLY	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	0.79 [0.75, 0.82]
Qwen3.5-9B	FULL	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	1.00 [1.00, 1.00]
	REC-ONLY	0.17 [0.13, 0.20]	0.03 [0.02, 0.05]	0.08 [0.05, 0.10]	0.03 [0.01, 0.04]	0.27 [0.23, 0.30]
	KV-ONLY	0.06 [0.04, 0.08]	0.00 [0.00, 0.01]	0.04 [0.03, 0.06]	0.02 [0.01, 0.03]	0.77 [0.73, 0.81]
Falcon-H1-3B	FULL	0.02 [0.00, 0.03]	0.00 [0.00, 0.01]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	1.00 [0.99, 1.00]
	REC-ONLY	0.15 [0.11, 0.20]	0.01 [0.00, 0.02]	0.06 [0.04, 0.09]	0.04 [0.02, 0.06]	0.30 [0.24, 0.35]
	KV-ONLY	0.01 [0.00, 0.02]	0.04 [0.02, 0.06]	0.00 [0.00, 0.00]	0.00 [0.00, 0.01]	0.65 [0.61, 0.69]
Falcon-H1-7B	FULL	0.01 [0.00, 0.02]	0.00 [0.00, 0.00]	0.00 [0.00, 0.01]	0.00 [0.00, 0.00]	1.00 [1.00, 1.00]
	REC-ONLY	0.75 [0.71, 0.80]	0.68 [0.63, 0.72]	0.61 [0.55, 0.66]	0.48 [0.42, 0.55]	0.96 [0.94, 0.98]
	KV-ONLY	0.09 [0.06, 0.11]	0.21 [0.18, 0.25]	0.08 [0.06, 0.12]	0.08 [0.06, 0.10]	0.82 [0.79, 0.85]
Qwen3.5-27B‡	FULL	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	1.00 [1.00, 1.00]
	REC-ONLY	0.52 [0.45, 0.59]	0.20 [0.15, 0.26]	0.50 [0.43, 0.57]	0.52 [0.45, 0.59]	0.59 [0.52, 0.66]
	KV-ONLY	0.31 [0.25, 0.38]	0.01 [0.00, 0.02]	0.30 [0.23, 0.36]	0.38 [0.31, 0.44]	0.98 [0.95, 1.00]
Falcon-H1-34B‡	FULL	0.01 [0.00, 0.02]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	1.00 [1.00, 1.00]
	REC-ONLY	0.15 [0.10, 0.20]	0.00 [0.00, 0.00]	0.05 [0.02, 0.08]	0.02 [0.01, 0.04]	0.39 [0.32, 0.45]
	KV-ONLY	0.17 [0.12, 0.22]	0.02 [0.00, 0.04]	0.12 [0.08, 0.17]	0.12 [0.08, 0.17]	0.68 [0.61, 0.74]

Table 7: Memory-conjunction trap (Demo 4). Five rates per (model, mode): conjunction-absent FPR (the trap), neutral-absent FPR (yes-bias control), left-only and right-only FPR (only one of the two morphemes present in the list), and present TPR (compound itself in the list). Unmarked rows: n=200 per condition per seed, 3 seeds, 95% bootstrap CIs. ‡Single-seed extension (seed 42 only); intervals are within-seed bootstraps.