# Preserving Memory Function in Hybrid Language Models: A Survey of State, Evidence, Access, and Granularity

**Anonymous ACL submission**

## Abstract

Hybrid language models combine Transformer attention with recurrent, state-space, or online-updated components to lower long-context serving cost while preserving memory fidelity. Existing surveys mostly organize this space by architecture family, chronology, or efficiency technique, leaving workload-specific memory functions underspecified. We propose a State-Evidence-Access (SEA) framework: tasks impose pressures on active latent state, addressable evidence, and access/update bandwidth, which architectures allocate at different granularities. This view explains why attention ratios or average perplexity can be misleading: a hybrid can save memory while losing the retrieval, state-update, or access function needed by the task. We survey hybrids by allocation locus—from key–value (K-V) and head-level mechanisms through layer, block, external-memory, and system-cache policies—and characterize their *function-granularity mismatch* risks. The result is a diagnostic map: in workloads such as long-document QA, does failure reflect missing evidence, unstable state updates, access bottlenecks, or poor integration?

## 1 Introduction

Long-context language models must retain relevant evidence at manageable inference cost. Transformer self-attention (Vaswani et al., 2017) keeps a key–value (K-V) store for direct token access, but KV-cache memory grows with length. State-space models (SSMs) and gated RNNs compress history into bounded active state (Gu and Dao, 2023), enabling streaming generation but weakening arbitrary copying and retrieval (Jelassi et al., 2024; Wen et al., 2025). State tracking adds pressure because common SSMs can share bounded-depth Transformer limits for reliable state evolution (Merrill et al., 2024). Hybridization is therefore not merely an efficiency compromise: no standard mechanism cleanly provides addressable token evidence, reliable compressed-state evolution, and bounded access/update cost.

Table 1: Positioning relative to neighboring survey lines.

| Survey line | Primary organizing emphasis |
|---|---|
| SSM/Mamba | Mechanism families and applications. |
| Long-context | Context-extension techniques and benchmarks. |
| Efficient attention / KV cache | Attention cost and serving memory. |
| Augmented LMs | External tools and retrieval augmentation. |
| This survey | Memory function / allocation locus under SEA budgets. |

Recent hybrids combine attention with recurrent, state-space, or online-updated components. Jamba (Lieber et al., 2024), Griffin (De et al., 2024), Zamba (Glorioso et al., 2024), Hymba (Dong et al., 2025), Nemotron-H (NVIDIA, 2025), and Falcon-H1 (Falcon LLM Team, 2025) span these regimes. Prior surveys catalogue SSM/Mamba families (Qu et al., 2024), long-context techniques (Liu et al., 2025; Wang et al., 2024b), efficient attention (Sun et al., 2025c), augmentation (Mialon et al., 2023), and KV-cache management (Li et al., 2025a). Our focus is complementary (Table 1): we ask which retrieval, update, or access function survives attention replacement, sparsification, or compression under state/evidence/access budgets. Appendix H gives the detailed comparison.

We treat hybrid language models as *function-preserving memory-allocation systems*. We propose the State-Evidence-Access (SEA) framework which separates active latent state, addressable evidence, and access/update bandwidth, then asks which memory function each resource supports. Failures arise when compressed state substitutes for addressable evidence, retained evidence cannot be accessed or integrated under budget, or capacity is assigned at a granularity that misses the workload's function. Architectures choose a mechanism, allocation granularity, and interven-

tion phase: K-V or head-level choices, layer/block interleaving, external stores, training-time conversion, and serving/cache policies. SEA identifies resource/function requirements, allocation granularity describes where capacity is spent, and mechanistic probes test where trained models implement the required function.

At the finest granularity, this becomes a K-V allocation question: what evidence remains addressable, and what is compressed, externalized, or evicted? In long-document QA, one error may reflect evicted answer evidence, unrecoverable compressed state, failed routing, or failed integration. Section 3.6 formalizes this diagnostic question; Appendix B records evidence use. In summary, our contributions are:

- **SEA framework.** We organize hybrid memory design around state, evidence, and access budgets, with memory functions as roles over those resources.
- **Multi-axis taxonomy.** We recast hybrids by resource role, mechanism, allocation granularity, and intervention phase, then use representative design patterns to track mismatch risks.
- **Function-preservation diagnostics.** We connect capability pressures to resource/function requirements, K-V allocation, and alignment objectives to detect when hybridization preserves or loses retrieval-, update-, or access-critical behavior.

**Organization.** Section 2 defines the scope and survey protocol. Section 3 introduces SEA resource budgets and the fine-grained K-V allocation question. Section 4 identifies the capability pressures that expose different memory requirements. Section 5 maps hybrid architectures by allocation locus, and Section 6 asks what function must survive replacement or distillation. Section 7 summarizes open problems and limitations.

## 2 Scope, Definitions, and Survey Methodology

**Scope and definitions.** This survey covers autoregressive hybrids: language models that combine attention with recurrent, state-space, linear-attention, online-updated, external-memory, or cache-policy components to change long-context memory allocation. A model is *hybrid* when multiple memory mechanisms or allocation policies are architecturally meaningful, not merely because it uses an optimized attention kernel. We use *SSM* broadly for structured recurrent state updates. *Memory function* is the workload role of a resource (lookup, state update, associative prediction, external access, or cache control); *allocation locus* is the unit assigned that role (K-V pair, head, layer, block, external store, or cache policy); *function-granularity mismatch* means the locus is too coarse or misaligned.

**Inclusion boundaries.** We include theory, mechanisms, benchmarks, conversion methods, and systems work that clarify retained evidence, compressed state, online associative update, external lookup, or cache/access policy for autoregressive text modeling. Broad multimodal, vision, graph, genomics, and time-series hybrids are boundary evidence, not catalogue targets.

**Survey protocol.** Our protocol is targeted rather than PRISMA-style exhaustive, but auditable. We searched ACL Anthology, arXiv, OpenReview, Semantic Scholar, and major conference/system proceedings for 2017–2026 work using keyword families around *hybrid language model*, *SSM/Mamba*, *linear/recurrent language model*, *long context*, *memory compression*, *retrieval head*, *KV cache*, *external memory*, and *Transformer-to-SSM/Mamba distillation*, then snowballed from related surveys and benchmarks. Each work is coded by resource role, mechanism, allocation granularity, intervention phase, memory function, mismatch risk, benchmark/probe type, and evidence status (peer-reviewed, preprint, technical report/blog, theory, benchmark, or mechanistic analysis). Appendices record the protocol and evidence-use tables; evidence status helps keep single recent reports from carrying strong claims.

## 3 Resource Budgets and the Fine-Grained K-V Problem

We start from a familiar contrast: attention keeps token evidence addressable; SSM/RNN mechanisms keep compressed state active; and deployment controls what can be read, routed, or updated per step. SEA names these pressures active latent state, addressable evidence, and access/update bandwidth. State and evidence are representational resources; access is the budget for reading, writing, routing, and integrating them. We write $B_t$ for active latent-state capacity, $S_t/M_t$ for retained or external addressable evidence, and $R_t$ for per-step

access/update budget. In practice, $B_t$ can be active recurrent-state or latent-slot bytes, $S_t/M_t$ retained K-V tokens/bytes or archive size, and $R_t$ per-token cache reads, routing or eviction calls, and memory bandwidth. Section 3.6 formalizes this allocation problem; Figure 1 summarizes the path from capability pressure to function, resource, locus, and mismatch.

The object-level contrast is Transformer episodic token memory versus SSM compressed relational memory. Transformer attention (Vaswani et al., 2017) and its KV cache keep past K-V pairs addressable by content query; SSM hidden state (Gu et al., 2022; Gu and Dao, 2023) fits patterns, transitions, and predictive relations in bounded latent state. A *K-V pair* is one token position's key and value vectors in an attention layer; retaining it keeps that position addressable rather than compressed.

We use "memory allocation" for how these budgets are spent: *latent state capacity*, the recurrent, SSM, or fast-weight variables active in the dynamics (Gu and Dao, 2023; Schlag et al., 2021); *addressable evidence capacity*, the KV cache, sparse/local cache, external store, or other lookup evidence; and *access/update bandwidth*, the per-step budget for reading, routing, updating, or integrating those resources. Growing external evidence is not a growing latent state: external evidence preserves items for lookup, but still costs retrieval and integration bandwidth. Table 2 summarizes this distinction as a resource-regime matrix for autoregressive inference.

**Historical anchors.** Earlier long-context lines already instantiate these regimes: Transformer-XL (Dai et al., 2019) and Compressive Transformer (Rae et al., 2020) place segment or compressed memories between retained evidence and compressed state; kNN-LM, RETRO, and Memorizing Transformers (Khandelwal et al., 2020; Borgeaud et al., 2022; Wu et al., 2022) externalize evidence; and RMT, Landmark Attention, LongNet, Hyena, and Mega (Bulatov et al., 2022; Mohtashami and Jaggi, 2023; Ding et al., 2023; Poli et al., 2023; Ma et al., 2023) change memory tokens, access patterns, or sequence operators. We use them as boundary evidence for the same state/evidence/access distinction.

### 3.1 Explicit Content-Addressable Retrieval Memory

Transformer self-attention implements explicit content-addressable evidence: each query scores retained keys and reads their values, giving token-level access to past evidence (Vaswani et al., 2017). Its computation and KV-cache storage grow with retained context. Sparse or local attention (Deng et al., 2024) reduces bandwidth by limiting which positions remain readable, thereby adding an evidence-selection policy.

### 3.2 Compressed-State Recurrent Memory

Structured SSMs and gated recurrent networks replace per-token KV with bounded latent state (Gu et al., 2022; Gu and Dao, 2023). This enables $O(1)$ streaming because the active representation does not grow with sequence length, but it makes history available through state updates rather than through addressable token evidence. Targeted copying and content-addressable retrieval probes therefore expose gaps not explained by next-token loss alone (Jelassi et al., 2024; Wen et al., 2025); RWKV and RetNet are representative variants (Peng et al., 2023; Sun et al., 2023).

### 3.3 Online-Updated Predictor / Associative Map

Some mechanisms make $B_t$ associative or predictive: linear attention and GLA write K-V summaries recurrently (Katharopoulos et al., 2020; Yang et al., 2024); DeltaNet, TTT-style methods, feature-map approximations, and product-key fast-weight memory add overwrite-like, gradient, or sparse associative updates (Yang et al., 2025; Sun et al., 2025b; Zhang et al., 2024a; Zhao and Jones, 2026). The readout applies an updated associative operator rather than reading a stored token, so K-V-pair memorization remains update-governed and capacity-limited.

### 3.4 External and Persistent Memory

Retrieval-Augmented Generation (RAG) (Mialon et al., 2023) and product-key memory (Zhao and Jones, 2026) externalize evidence beyond parameters or activations. This can make evidence capacity much larger than active state, but usable memory remains bounded by retrieval and integration bandwidth. Engram (Cheng et al., 2026) adds deterministic lookup as a boundary case, extending SEA beyond SSM-attention hybrids.

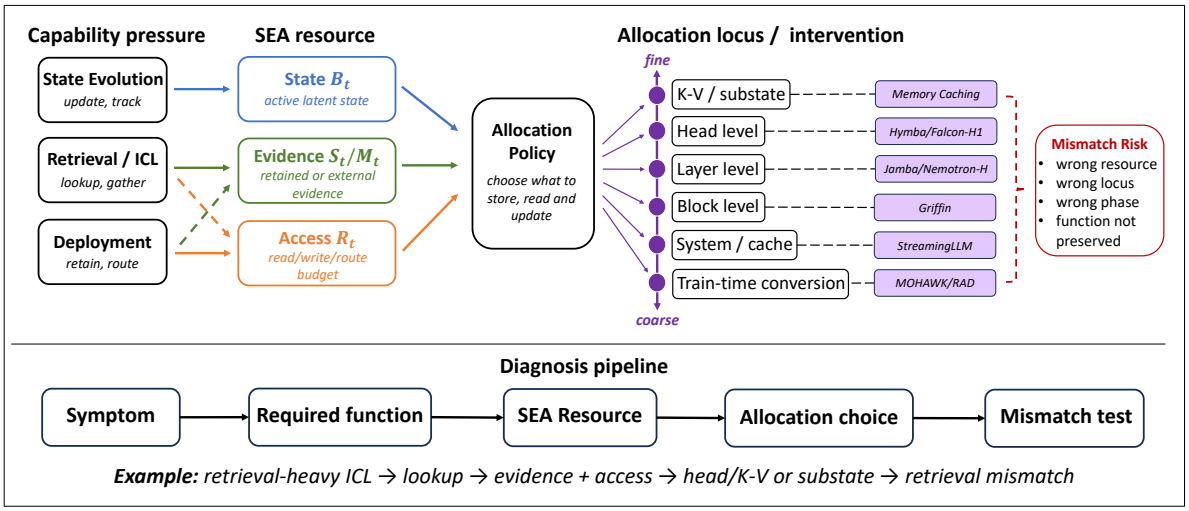


Figure 1: Diagnostic workflow for function-preserving allocation. Capability pressures map to memory functions, SEA resources, and allocation loci or cross-cutting interventions. Solid arrows mark dominant pressures; dashed arrows mark secondary couplings. Mismatch arises when the chosen locus or intervention is too coarse or misaligned for the function.

Table 2: Resource regimes in autoregressive inference.

| Regime | State $B_t$ | Evidence $S_t/M_t$ | Access $R_t$ | Consequence |
|---|---|---|---|---|
| Transformer | $O(1)$ query | $S_t = O(n)$ | $O(n)$ read | Exact evidence; growing cache/access. |
| Fixed SSM/RNN | $O(1)$ | $S_t = 0$ | $O(1)$ | Cheap streaming; compressed history. |
| Linear/TTT/fast weights | predictor/matrix | implicit | $O(1)$ update | Adaptive state; bounded capacity. |
| Local + recurrence | $O(1)$ global | $S_t = O(w)$ | $O(w) + O(1)$ | Exact window; compressed far tail. |
| Head/layer hybrid | mixed | selected $S_t$ | structured | Attention must match function. |
| External/lookup | small | $M_t \gg S_t$ | retrieval-limited | Scaled storage; retrieval bottleneck. |
| Growing/checkpoint | $O(g(n))$ | optional $S_t$ | adaptive | Between recurrence and full KV. |

### 3.5 Theoretical Unity: Structured State Space Duality

Dao and Gu (2024) express attention and SSMs as structured semiring matrix products, or Structured State Space Duality (SSD). This places them in one allocation space without making them behaviorally equivalent: the gaps in Section 4 arise from matrix structure, state size, and access constraints, not formal incompatibility.

### 3.6 The Fine-Grained K-V Allocation Lens

The resource-budget distinction raises a sharper question: at K-V or sub-head granularity, which evidence remains addressable, compressed into state, externalized, or ignored? We frame this as a compact formal lens, not a solved optimization problem. Its operational role is to define measurable intervention targets: query families, evidence units, resource budgets, access policies, and failure attribution.

Let the processed K-V evidence at time $t$ be

$$H_t = \{(k_i, v_i)\}_{i \le t}, \qquad q \sim \mathcal{Q}_t,$$

where $q$ is a future query drawn from the task-induced query distribution. The model has three resources: an active latent state $h_t \in \mathbb{R}^{B_t}$, a K-V retention cache

$$C_t = \{(k_i, v_i) : \pi_i = 1\}, \qquad |C_t| \le S_t,$$

and an external or persistent memory $M_t$. Let $\mathcal{R}_t = (h_t, C_t, M_t)$ denote the available resources. The allocation decisions are K-V retention indicators $\pi_i \in \{0, 1\}$, compressed or external updates $u_i$, and a routing/access policy $\rho(q)$ that decides which resources a query may read.

For a full-attention target, define

$$r^\star(q, H_t) = \sum_{i \le t} \alpha_i(q, H_t) v_i,$$
$$\hat{r}_\rho(q; \mathcal{R}_t) = A_{\rho(q)}(q; \mathcal{R}_t),$$

where $A_{\rho(q)}$ is the retrieval procedure induced by the chosen route. The budgeted allocation problem is then

$$\min_{\pi, u, \rho} \quad \mathbb{E}_{q \sim \mathcal{Q}_t} \left[\ell(r^\star(q, H_t), \hat{r}_\rho(q; \mathcal{R}_t))\right]$$
$$\text{s.t.} \quad |C_t| \le S_t, \dim(h_t) \le B_t, \mathrm{cost}(\rho, u) \le R_t.$$

The loss may measure attention-output recovery, target-value recovery, or downstream prediction error; the point is to expose the tradeoff between latent-state capacity, addressable evidence, and access/update bandwidth. In practice, $\mathrm{cost}(\rho, u)$ can count cache reads, routing or eviction calls, external lookups, and memory-bandwidth or latency budgets. The full-attention target is a diagnostic reference for addressable evidence, not a universal gold standard; state-evolution tasks may require task losses or state-transition objectives instead.

The variables are bookkeeping rather than a new training objective: start from a workload-induced query family $\mathcal{Q}_t$, choose a probe or intervention, estimate which K-V entries, heads, or layers are function-sensitive, then test whether failure comes from missing evidence, state-update loss, access limits, or poor integration. Copying and associative recall point to retained evidence; state tracking to compressed-state updates; and long-context serving to cache, routing, and integration policies.

Common architectures become special cases. A full Transformer uses $S_t = O(t)$, $B_t = O(1)$, and dense $R_t = O(t)$ access; a fixed SSM or RNN sets $S_t = 0$, $B_t = O(1)$, and $R_t = O(1)$; sparse attention selects an evidence subset; external memory enlarges $M_t$ but remains bounded by $R_t$; and growing-latent-state regimes scale $B_t = O(g(t))$ with $1 \ll g(t) \ll t$. Memory Caching (Behrouz et al., 2026) caches recurrent checkpoints and merges them with online state, illustrating state growth without literal token-level K-V selection. Hybrid Associative Memories (Lufkin et al., 2026) and fast-weight product-key memory (Zhao and Jones, 2026) are adjacent fixed-policy neighbors rather than settled allocation solutions.

**Operational diagnostic: long-document QA under a KV budget.** In long-document QA, $\mathcal{Q}_t$ contains answer-seeking queries that arrive after many distractors; $S_t$ is the retained KV budget, $M_t$ an optional chunk index, $B_t$ active recurrent/latent state, and $R_t$ the cache or external reads available per token. The K-V lens asks which answer-bearing or bridge evidence is sensitive to eviction, compression, quantization, or externalization. If the answer span is gone, the intervention is evidence retention; if it is compressed but not recoverable, state capacity or update objectives are implicated; if it is stored but not read, the access policy failed; and if it is read but not used, integration or routing failed. Thus the same accuracy drop points to different design changes depending on which SEA resource failed.

## 4 Capability Pressure: What Requirements Do Tasks Expose?

Resource budgets reveal *capability pressures*: task or deployment demands on state, evidence, and access. We group them into three clusters and map each to requirements, evidence, and architectural implications in Appendix D. NLP workloads instantiate the map: long-document QA stresses distant lookup and provenance; multi-document summarization stresses evidence selection and cross-document integration; dialogue, code, and agent traces stress state updates for constraints and bindings; retrieval-heavy ICL stresses associative recall; and multilingual long context stresses tokenization-sensitive evidence and state.

### 4.1 Three Pressure Clusters

Context-access tasks require a model to recover a specific token, binding, or pattern from the prefix. Copying lower bounds (Jelassi et al., 2024), in-context retrieval analysis (Wen et al., 2025), and broader subquadratic limitations (Alman and Yu, 2025) point to one requirement: some workloads need direct access to retained token-level evidence, not only bounded compressed state. Context-dependent sparse attention, as in HAX/CDSA (Zhan et al., 2025), is a positive example: it keeps access subquadratic by selecting evidence conditional on the query/context rather than through a fixed sparse pattern. The ICL picture is nuanced—Mamba-like models can show ICL-like behavior on smooth or low-dimensional regimes (Grazzi et al., 2024; Park et al., 2024)—but exact copying, associative recall, and retrieval-heavy ICL remain the stress points that motivate retaining some attention or K-V retention capacity.

State-evolution tasks instead require reliable latent updates across operations. This is not a blanket weakness of recurrence: gated nonlinear RNNs are designed for such updates. The negative evidence is narrower: common SSMs, linear recurrent variants, or constrained state-space parameterizations can impose update-rule limits distinct from retrieval failure (Merrill et al., 2024; Grazzi et al., 2025). Thus attention may supply retrieval anchors while state tracking still requires stable recurrent updates or finer sub-layer coordination.

Streaming and long-context serving add a

Table 3: Four axes for hybrid memory allocation.

| Axis | Values; diagnostic |
|---|---|
| Resource role | Active state; addressable evidence; access/update bandwidth. *Diagnostic:* what preserves the function? |
| Mechanism | Attention; recurrence/SSM; linear attention/fast weights; external memory; cache policy. *Diagnostic:* how is history stored or read? |
| Granularity | K-V/sub-head; head; layer; block/module; external store; system policy. *Diagnostic:* is the unit fine enough? |
| Phase | Pretraining; conversion/distillation; inference routing; serving/cache management. *Diagnostic:* when is the choice imposed? |

deployment-driven capability pressure: full attention preserves addressable evidence but pays KV-cache capacity and access-bandwidth costs, whereas recurrent paths provide fixed-state streaming at the price of information loss. The resulting requirement is a budget/control policy over what remains cache-retained, compressed, externalized, or evicted. The main-text mapping is: context access requires addressable evidence; state evolution requires reliable compressed updates; smooth ICL-like regimes motivate online-updated predictors; and deployment imposes budget policies over K-V retention, state, and access bandwidth. In the long-document QA example, the same map separates answer-span lookup, document-state tracking, and retrieval/integration bandwidth rather than treating failure as a generic long-context deficit.

## 5 Architecture Spectrum: Multi-Axis Hybrid Design

Given these conflicting resource demands, hybrid models are best understood as *allocation architectures*: they assign state/evidence/access resources and their functional roles at different granularities, from K-V operations and heads through layers, blocks, external memory, and cache policies. The patterns below can be read as increasingly coarse approximations to the K-V allocation problem: they choose where future-sensitive evidence remains addressable, compressed, routed, or evicted. To avoid conflating different kinds of design choices, we separate four axes (Table 3). Resource role asks what is being protected; mechanism asks how it is represented or accessed; granularity asks where capacity is assigned; and phase asks when the intervention occurs. Appendix E summarizes representative design patterns and citations.

**K-V / sub-head granularity.** At the finest granularity, the decision unit is the K-V pair or sub-head channel: which evidence remains addressable rather than summarized? Current hybrids mostly allocate at coarser head, layer, or block loci, but Memory Caching, hybrid associative memory, and fast-weight product-key memory point toward finer choices (Behrouz et al., 2026; Lufkin et al., 2026; Zhao and Jones, 2026). These are not literal K-V-pair solvers: cached checkpoints or associative updates may contain useful history yet omit the evidence a later query needs.

**Layer-wise interleaving.** The common strategy inserts full-attention anchors into a mostly recurrent stack so accumulated context-access demands can be resolved and passed downstream. Jamba (Lieber et al., 2024), Samba (Ren et al., 2025), and Nemotron-H (NVIDIA, 2025) use different mixes of attention, recurrence, and gated linear attention. Because attention ratio and placement affect capabilities (Bae et al., 2025), and ICL functions can concentrate unevenly (Wang et al., 2025b), placement is the allocation variable, not ratio alone.

**Intra-layer and head-level fusion.** A finer strategy assigns heads or channels inside a block to different memory mechanisms. Hymba (Dong et al., 2025) and Falcon-H1 (Falcon LLM Team, 2025) mix attention-like and recurrent channels; mechanistic work suggests retrieval-critical heads can preserve retained evidence while other heads carry compressed dynamics or local refinement (Bick et al., 2025b; Michalak and Abreu, 2025). The split helps only when protected heads align with retrieval- or update-critical functions.

**Shared attention modules.** Shared attention routes multiple positions through one centralized retrieval module. Zamba (Glorioso et al., 2024) lowers KV-cache overhead this way, but one module may not serve every layer-local retrieval need.

**Local attention plus global recurrence.** Local attention plus global recurrence gives direct access to a short recent window and compresses the tail into recurrent state. Griffin (De et al., 2024) and RecurrentGemma (Botev et al., 2024) instantiate this division. The design is attractive for state evolution but fragile when far-past evidence becomes retrieval-critical and cannot be reconstructed from state.

Table 4: Representative systems by allocation handle, publication status, and failure mode (pub.=published; pre.=preprint/report).

| System line | Allocation handle | Pub. | Failure mode |
|---|---|---|---|
| Jamba; Jamba-1.5 | Layer anchors in Mamba-heavy stack | pre. | Misplaced retrieval |
| Griffin; RecurrentGemma | Local KV + recurrent global state | pre. | Far-tail loss |
| Zamba | Shared attention; amortized KV | pre. | Coverage gaps |
| Hymba; Falcon-H1 | Head/block fusion; attn.+recur. channels | pub./pre. | Head/function gap |
| Samba; Nemotron-H | Production mix; attn.+recur.+linear | pub./pre. | Claim strength |
| Attn.–Mamba conversion | Replace/linearize attention | pub./pre. | Output/function gap |
| StreamingLLM; InfLLM | Cache sinks/recent tokens; evicted-block retrieval | pub. | Evicted evidence |
| DuoAttention; SnapKV | Head/prompt-token KV selection | pub. | Future-relevance miss |
| vLLM; H2O; KIVI | Paging; eviction; KV quantization | pub. | Byte/function gap |

**Dynamic switching and adaptive routing.** Dynamic routing selects attention or recurrence by input or position. TransMamba (Li et al., 2025b) and TransXSSM (Wu et al., 2025a) approximate the decision in Section 3.6: spend attention where future relevance justifies it. The router must predict future retrieval relevance before spending the comparison budget it is trying to save; routing stability, supervision, and cheap content comparison remain open (Basu, 2026).

**Training-time allocation and conversion.** Distillation and conversion *compile* a teacher's retrieval behavior into recurrent, linear-attention, or hybrid weights. We treat them as intervention-phase choices: the risk is preserving average predictions while losing teacher components that carry retrieval, state-update, or access behavior.

**System-level and cache allocation.** At deployment time, KV capacity, recurrent-state size, external evidence, access bandwidth, and cache policy decide which memory functions remain usable per token. StreamingLLM retains attention sinks (Xiao et al., 2024b); InfLLM retrieves evicted blocks (Xiao et al., 2024a); DuoAttention uses head-aware budgets (Xiao et al., 2025); FlashAttention changes bandwidth constants (Dao et al., 2022; Dao, 2024); and vLLM/H2O/SnapKV/KIVI provide paging, eviction, selection, or quantization (Kwon et al., 2023; Zhang et al., 2023; Li et al., 2024a; Liu et al., 2024). SideQuest (Kariyappa and Suh, 2026) illustrates the failure mode: cache savings are valid only if multi-hop reasoning evidence remains readable.

## 6 Function-Granularity Mismatch: What Must Be Preserved?

Section 5 assigns resources at practical loci such as heads, layers, blocks, and cache policies. The diagnostic question is whether that locus preserves the needed function. A model can preserve perplexity while losing copying, associative recall, or retrieval-heavy ICL if remaining attention misses retrieval-carrying components. G&A heads are the clearest mechanistic case below, but state-update and access-policy failures follow the same SEA diagnosis: distant evidence becomes unreachable, latent state drifts, or stored evidence cannot be read under budget.

**Three mismatch modes.** Retrieval mismatch asks whether needed evidence remains addressable; state-update mismatch asks whether compressed dynamics maintain the latent variables a task needs; and access-policy mismatch asks whether retained or external evidence can be read and integrated under budget. We detail retrieval because current mechanistic evidence is strongest at the head/layer level, while treating state-update and access-policy failures as parallel diagnostics for SSM updates and cache policies.

**Retrieval-critical components.** Mechanistic studies show that attention heads need not serve the same function. Bick et al. (2025b) identify *Gather-and-Aggregate* (G&A) heads for in-context retrieval: they gather relevant signals and route them for downstream aggregation. Wu et al. (2025b) similarly characterize retrieval-specialized heads by factuality and information routing. This is distinct from heads that compute local patterns, aggregate nearby representations, or serve as positional anchors. The lesson for Section 4 is *function-granularity mismatch*: replacing, distilling, or caching across components that implement addressable-evidence access can preserve attention budget or perplexity while losing the learned retrieval function. Adjacent failures are state-update mismatch, where the recurrent/SSM update cannot maintain required latent state, and access-policy mismatch, where

evidence is stored but not readable.

**Function vs. perplexity.** Perplexity matching does not imply function preservation: a recurrent or hybrid model can match language-modeling loss while degrading retrieval on context-access tasks (Bick et al., 2025b). Architecture evaluations of ICL and Transformer/SSM behavior (Wang et al., 2025b; Arora et al., 2025) reinforce that aggregate quality can hide different routes to context use. Replacing attention with recurrence should therefore preserve or replace G&A-type retrieval behavior, either in remaining attention or an explicit retrieval mechanism.

**Alignment targets for distillation.** Distillation and conversion choose which teacher behaviors, hidden states, attention maps, or retrieval components to compile into a cheaper recurrent, linear-attention, or hybrid student. MOHAWK (Bick et al., 2024), Goldstein et al. (Goldstein et al., 2025), Mamba-in-Llama (Wang et al., 2024a), LoLCATs (Zhang et al., 2025), Attention Bridge (Wang et al., 2025a), and Priming (Chattopadhyay et al., 2026) illustrate weight reuse, attention linearization, representation alignment, and layer transfer. In Priming, for example, selected Transformer attention layers become SSM layers, source weights transfer, and each new SSM layer is aligned before end-to-end adaptation. The central choice is the alignment target: logits or hidden states do not guarantee learned retrieval, state, or access roles. RAD-style retrieval-aware distillation (Bick et al., 2026) targets context-access behavior directly, while Llamba (Bick et al., 2025a) supports scalable cross-architecture distillation without explicitly preserving G&A heads.

**How much attention is needed?** A practical consequence is *minimal attention*: how many layers or retrieval heads preserve context-access capability? Empirical evidence suggests a few specialized components can carry disproportionate retrieval behavior (Michalak and Abreu, 2025), and targeted full attention can protect retrieval when many layers are linearized (Goldstein et al., 2025). The design principle is to identify which components carry retrieval function and allocate attention there, not to preserve a fixed attention ratio.

## 7 Open Problems

**Function-aware and adaptive allocation.** Theory should turn the K-V objective in Section 3.6 into head-, layer-, block-, and system-level policies: what stays addressable, what is compressed, and what adapts online? SSD (Dao and Gu, 2024) unifies attention and SSMs algebraically, but not by allocation granularity, and Memory Caching (Behrouz et al., 2026) is an early boundary case rather than a full policy. Conversion and distillation likewise need functional targets, not just logits or hidden states. Future controllers may mix fixed $O(1)$ recurrence, sublinear active state, and sparse retained evidence; the key question is when they can replace direct $O(n)$ evidence access without losing independent bindings.

**Evaluation and reporting.** Benchmarks often merge copying, associative recall, retrieval-heavy ICL, state tracking, and long-context lookup into aggregate scores. MQAR/Zoology (Arora et al., 2024a), copying lower-bounds (Jelassi et al., 2024), LongBench (Bai et al., 2023), and SSM/hybrid evaluations (Mitra et al., 2025) separate pieces, but a unified protocol should map failures to SEA pressures and allocation loci. It should also cover multilingual and safety-critical memory, where tokenization, provenance, constraints, and sensitive data can change which history must remain addressable.

**Systems co-design.** KV-cache paging, eviction, quantization, sparse kernels, and recurrent state layouts determine feasible $R_t$ and $S_t$ budgets. Hybrid papers should report memory-function tradeoffs together with throughput and memory numbers, so systems gains can be separated from retrieval or state-update loss.

## 8 Conclusion

Hybrid design should preserve workload-required memory functions, not only average efficiency or output quality. SEA resources, the K-V lens, and multi-axis design patterns expose where heuristics fail through function-granularity mismatch: the central question is which memory function remains addressable under a given budget, not only how much attention remains.

Progress therefore requires moving from architecture-level heuristics to function-preserving memory allocation: identify the workload's retrieval, state-update, and access needs; allocate the right resource at the right granularity; and evaluate the function directly rather than relying only on aggregate language-modeling quality.

## Limitations

This survey is a targeted synthesis of autoregressive language-model memory mechanisms, not a full systematic review or exhaustive model catalogue. Vision, audio, and multimodal hybrids are treated only as boundary cases. Several recent systems and benchmark analyses are preprints or technical reports; we use them as evidence for emerging patterns rather than as conclusive quantitative support. Formal and mechanistic claims are interpreted under their source assumptions.

**Potential risks.** This survey introduces no new model, dataset, or deployment system, so its direct risks are limited. Indirectly, more efficient long-context and hybrid models may lower the cost of large-scale generation, retrieval, surveillance-scale document analysis, and agentic tool use. Persistent memory, external retrieval, and KV retention may also keep sensitive evidence available longer than users expect, while cache compression or eviction can remove safety-relevant instructions, constraints, provenance, or consent information. Deployment therefore requires task-specific safety, privacy, and misuse evaluation.

Our protocol is auditable but not independently multi-annotated, and long-context benchmarks remain English-heavy and often synthetic. We treat benchmark results as pressure evidence rather than complete coverage of multilingual or low-resource memory behavior. "Retrieval" is used in the broad memory-systems sense of content-addressable access, not as a claim about information-retrieval system performance.

## References


Josh Alman and Hantao Yu. 2025. Fundamental limitations on subquadratic alternatives to transformers. In *International Conference on Learning Representations*.

Aryaman Arora, Neil Rathi, Nikil Roashan Selvam, Róbert Csordás, Dan Jurafsky, and Christopher Potts. 2025. Mechanistic evaluation of transformers and state space models. *arXiv preprint arXiv:2505.15105*.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. 2024a. Zoology: Measuring and improving recall in efficient language models. *arXiv preprint arXiv:2312.04927*.

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, Dylan Zinsley, James Zou, Atri Rudra, and Christopher Ré. 2024b. Simple linear attention language models balance the recall-throughput tradeoff. *arXiv preprint arXiv:2402.18668*.

Simran Arora, Aman Timalsina, Aaryan Singhal, Benjamin Spector, Sabri Eyuboglu, Xinyi Zhao, Ashish Rao, Atri Rudra, and Christopher Ré. 2024c. Just read twice: Closing the recall gap for recurrent language models. *arXiv preprint arXiv:2407.05483*.

Sangmin Bae, Bilge Acun, Chien-Yu Lin, Haroun Habeeb, Seungyeon Kim, Liang Luo, Junjie Wang, and Carole-Jean Wu. 2025. Hybrid architectures for language models: Systematic analysis and design insights. *arXiv preprint arXiv:2510.04800*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. LongBench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Abhinaba Basu. 2026. When does content-based routing work? representation requirements for selective attention in hybrid sequence models. *arXiv preprint arXiv:2603.20997*.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. 2024. xLSTM: Extended long short-term memory. *arXiv preprint arXiv:2405.04517*.

Ali Behrouz, Zeman Li, Yuan Deng, Peilin Zhong, Meisam Razaviyayn, and Vahab Mirrokni. 2026. Memory caching: RNNs with growing memory. *arXiv preprint arXiv:2602.24281*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Aviv Bick, Tobias Katsch, Nimit Sohoni, Arjun Desai, and Albert Gu. 2025a. Llamba: Scaling distilled recurrent models for efficient language processing. *arXiv preprint arXiv:2502.14458*.

Aviv Bick, Kevin Y. Li, Eric P. Xing, J. Zico Kolter, and Albert Gu. 2024. Transformers to SSMs: Distilling quadratic knowledge to subquadratic models. In *Advances in Neural Information Processing Systems*.

Aviv Bick, Eric P. Xing, and Albert Gu. 2025b. Understanding the skill gap in recurrent language models: The role of the Gather-and-Aggregate mechanism. In *International Conference on Machine Learning*.

Aviv Bick, Eric P. Xing, and Albert Gu. 2026. Retrieval-aware distillation for Transformer-SSM hybrids. *arXiv preprint arXiv:2602.11374*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, and 1 others. 2022. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning*.

Aleksandar Botev, Soham De, Samuel L. Smith, Anushan Fernando, George-Cristian Muraru, Ruba Haroun, and 1 others. 2024. RecurrentGemma: Moving past transformers for efficient open language models. *arXiv preprint arXiv:2404.07839*.

Aydar Bulatov, Yuri Kuratov, and Mikhail Burtsev. 2022. Recurrent memory transformer. *arXiv preprint arXiv:2207.06881*.

Loïc Cabannes, Maximilian Beck, Gergely Szilvasy, Matthijs Douze, Maria Lomeli, Jade Copet, Pierre-Emmanuel Mazaré, Gabriel Synnaeve, and Hervé Jégou. 2025. Short window attention enables long-term memorization. *arXiv preprint arXiv:2509.24552*.

Aditya Chattopadhyay, Elvis Nunez, Prannay Kaul, Benjamin Bowman, Evan Becker, Luca Zancato, David Thomas, Wei Xia, and Stefano Soatto. 2026. Priming: Hybrid state space models from pre-trained transformers. *arXiv preprint arXiv:2605.08301*.

An Chen and 1 others. 2023. L-Eval: Instituting standardized evaluation for long context language models. *arXiv preprint arXiv:2307.11088*.

Yingfa Chen, Zhen Leng Thai, Zihan Zhou, Zhu Zhang, Xingyu Shen, Shuo Wang, Chaojun Xiao, Xu Han, and Zhiyuan Liu. 2026. Hybrid linear attention done right: Efficient distillation and effective architectures for extremely long contexts. *arXiv preprint arXiv:2601.22156*.

Xin Cheng, Wangding Zeng, Damai Dai, Qinyu Chen, Bingxuan Wang, Zhenda Xie, Kezhao Huang, Xingkai Yu, Zhewen Hao, Yukun Li, Han Zhang, Huishuai Zhang, Dongyan Zhao, and Wenfeng Liang. 2026. Conditional memory via scalable lookup: A new axis of sparsity for large language models. *arXiv preprint arXiv:2601.07372*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Nicola Muca Cirone, Antonio Orvieto, Benjamin Walker, Cristopher Salvi, and Terry Lyons. 2024. Theoretical foundations of deep selective state-space models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988. Association for Computational Linguistics.

Tri Dao. 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35.

Tri Dao and Albert Gu. 2024. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning*.

Soham De, Samuel L. Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, Guillaume Desjardins, Arnaud Doucet, David Budden, Yee Whye Teh, Razvan Pascanu, Nando De Freitas, and Caglar Gulcehre. 2024. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*.

Yichuan Deng, Zhao Song, Jing Xiong, and Chiwun Yang. 2024. How sparse attention approximates exact attention? your attention is naturally $n^c$ sparse. *arXiv preprint arXiv:2404.02690*.

Jiayu Ding and 1 others. 2023. LongNet: Scaling transformers to 1,000,000,000 tokens. *arXiv preprint arXiv:2307.02486*.

Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, Zijia Chen, Ameya Sunil Mahabaleshwarkar, Shih-Yang Liu, Matthijs Van Keirsbilck, Min-Hung Chen, Yoshi Nishi, Yingyan Celine Lin, Jan Kautz, and Pavlo Molchanov. 2025. Hymba: A hybrid-head architecture for small language models. In *International Conference on Learning Representations*.

Igor Dubinin, Antonio Orvieto, and Felix Effenberger. 2026. Improved state mixing in higher-order and block diagonal linear recurrent networks. *arXiv preprint arXiv:2602.12021*.

Falcon LLM Team. 2025. Falcon-H1: A family of hybrid-head language models redefining efficiency and performance. *arXiv preprint arXiv:2507.22448*.

Parsa Ashrafi Fashi, Utkarsh Saxena, Mehdi Rezagholizadeh, Aref Jafari, Akash Haridas, Mingyu Yang, Vansh Bhatia, Guihong Li, Vikram Appia, and Emad Barsoum. 2026. Long-context aware upcycling: A new frontier for hybrid LLM scaling. *arXiv preprint arXiv:2604.24715*.

Alexander M. Fichtl, Jeremias Bohn, Josefin Kelber, Edoardo Mosca, and Georg Groh. 2025. The end of transformers? on challenging attention and the rise of sub-quadratic architectures. *arXiv preprint arXiv:2510.05364*.

Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. 2023. Hungry hungry hippos: Towards language modeling with state space models. In *International Conference on Learning Representations (ICLR)*.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge. 2024. Zamba: A compact 7B SSM hybrid model. *arXiv preprint arXiv:2405.16712*.

Daniel Goldstein, Eric Alcaide, Janna Lu, and Eugene Cheah. 2025. RADLADS: Rapid attention distillation to linear attention decoders at scale. In *Conference on Language Modeling*.

Riccardo Grazzi, Julien Siems, Simon Schrodi, Thomas Brox, and Frank Hutter. 2024. Is mamba capable of in-context learning? *arXiv preprint arXiv:2402.03170*.

Riccardo Grazzi, Julien Siems, Arber Zela, Jörg K.H. Franke, Frank Hutter, and Massimiliano Pontil. 2025. Unlocking state-tracking in linear RNNs through negative eigenvalues. In *International Conference on Learning Representations*.

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

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. 2020. HiPPO: Recurrent memory with optimal polynomial projections. In *Advances in Neural Information Processing Systems*, volume 33.

Albert Gu, Karan Goel, and Christopher Ré. 2022. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*.

Yuxian Gu, Qinghao Hu, Shang Yang, Haocheng Xi, Junyu Chen, Song Han, and Han Cai. 2025. Jet-Nemotron: Efficient language model with post neural architecture search. In *Advances in Neural Information Processing Systems*.

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

Jamba Team. 2024. Jamba-1.5: Hybrid transformer-Mamba models at scale. *arXiv preprint arXiv:2408.07887*.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. 2024. Repeat after me: Transformers are better than state space models at copying. In *International Conference on Machine Learning*.

Sanjay Kariyappa and G. Edward Suh. 2026. SideQuest: Model-driven KV cache management for long-horizon agentic reasoning. *arXiv preprint arXiv:2602.22603*.

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 (ICML)*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM Symposium on Operating Systems Principles*.

Disen Lan, Weigao Sun, Jiaxi Hu, Jusen Du, and Yu Cheng. 2025. Liger: Linearizing large language models to gated recurrent structures. *arXiv preprint arXiv:2503.01496*.

Haoyang Li, Yiming Li, Anxin Tian, Tianhao Tang, Zhanchao Xu, Xuejia Chen, Nicole Hu, Wei Dong, Qing Li, and Lei Chen. 2025a. A survey on large language model acceleration based on KV cache management. *Transactions on Machine Learning Research*.

Yixing Li, Ruobing Xie, Zhen Yang, Xingwu Sun, Shuaipeng Li, Weidong Han, Zhanhui Kang, Yu Cheng, Chengzhong Xu, Di Wang, and Jie Jiang. 2025b. TransMamba: Flexibly switching between transformer and mamba. In *AAAI Conference on Artificial Intelligence*.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024a. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024b. EAGLE: Speculative sampling requires rethinking feature uncertainty. In *International Conference on Machine Learning (ICML)*.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, and 3 others. 2024. Jamba: A hybrid Transformer-Mamba language model. *arXiv preprint arXiv:2403.19887*.

Ling Team. 2025. Every attention matters: An efficient hybrid architecture for long-context reasoning. *arXiv preprint arXiv:2510.19338*.

Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, Yuanxing Zhang, Zhuo Chen, Hangyu Guo, Shilong Li, Ziqiang Liu, Yong Shan, Yifan Song, Jiayi Tian, Wenhao Wu, and 3 others. 2025. A comprehensive survey on long context language modeling. *arXiv preprint arXiv:2503.17407*.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *International Conference on Machine Learning*.

Leon Lufkin, Tomas Figliolia, Beren Millidge, and Kamesh Krishnamurthy. 2026. Hybrid associative memories. *arXiv preprint arXiv:2603.22325*.

Xuezhe Ma, Chunting Zhou, Xiang Kong, Junxian He, Liangke Gui, Graham Neubig, Jonathan May, and Luke Zettlemoyer. 2023. MEGA: Moving average equipped gated attention. In *International Conference on Learning Representations*.

William Merrill, Jackson Petty, and Ashish Sabharwal. 2024. The illusion of state in state-space models. In *International Conference on Machine Learning*.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: A survey. *arXiv preprint arXiv:2302.07842*.

Felix Michalak and Steven Abreu. 2025. Some attention is all you need for retrieval. *arXiv preprint arXiv:2510.19861*.

MiniCPM Team. 2026. MiniCPM-SALA: Hybridizing sparse and linear attention for efficient long-context modeling. *arXiv preprint arXiv:2602.11761*.

MiniMax Team. 2025. MiniMax-01: Scaling foundation models with lightning attention. *arXiv preprint arXiv:2501.08311*.

Saptarshi Mitra, Rachid Karami, Haocheng Xu, Sitao Huang, and Hyoukjun Kwon. 2025. Characterizing SSM and hybrid performance with long context. *arXiv preprint arXiv:2507.12442*.

Amirkeivan Mohtashami and Martin Jaggi. 2023. Landmark attention: Random-access infinite context length for transformers. *arXiv preprint arXiv:2305.16300*.

Piotr Nawrot, Robert Li, Renjie Huang, Sebastian Ruder, Kelly Marchisio, and Edoardo M. Ponti. 2025. The sparse frontier: Sparse attention trade-offs in Transformer LLMs. *arXiv preprint arXiv:2504.17768*.

Chien Van Nguyen, Huy Nguyen, Ruiyi Zhang, Hanieh Deilamsalehy, Puneet Mathur, Viet Lai, Haoliang Wang, Jayakumar Subramanian, Ryan A. Rossi, Trung Bui, Nikos Vlassis, Franck Dernoncourt, and Thien Huu Nguyen. 2025. Lizard: An efficient linearization framework for large language models. *arXiv preprint arXiv:2507.09025*.

NVIDIA. 2025. Nemotron-H: A family of accurate and efficient hybrid Mamba-Transformer models. *arXiv preprint arXiv:2504.03624*.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, and 1 others. 2022. In-context learning and induction heads. *Transformer Circuits Thread*.

Adnan Oomerjee, Zafeirios Fountas, Haitham Bou-Ammar, and Jun Wang. 2026. Bottlenecked transformers: Periodic KV cache consolidation for generalised reasoning. In *International Conference on Learning Representations*.

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 comparisons study of in-context learning tasks. In *International Conference on Machine Learning*.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, and 13 others. 2023. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14048–14077.

Bo Peng, Daniel Goldstein, Quentin Anthony, Alon Albalak, Eric Alcaide, Stella Biderman, and 1 others. 2024. Eagle and finch: RWKV with matrix-valued states and dynamic recurrence. In *International Conference on Machine Learning (ICML)*.

Bo Peng, Ruichong Zhang, Daniel Goldstein, Eric Alcaide, Xingjian Du, Haowen Hou, Jiaju Lin, Jiaxing Liu, Janna Lu, William Merrill, Guangyu Song, Kaifeng Tan, Saiteja Utpala, Nathan Wilce, Johan S. Wind, Tianyi Wu, Daniel Wuttke, and Christian Zhou-Zheng. 2025. RWKV-7 "goose" with expressive dynamic state evolution. *arXiv preprint arXiv:2503.14456*.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. 2023. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*.

Zhen Qin, Songlin Yang, Weixuan Sun, Xuyang Shen, Dong Li, Weigao Sun, and Yiran Zhong. 2024. HGRN2: Gated linear RNNs with state expansion. In *Conference on Language Modeling*.

Haohao Qu, Liangbo Ning, Rui An, Wenqi Fan, Tyler Derr, Hui Liu, Xin Xu, and Qing Li. 2024. A survey of Mamba. *arXiv preprint arXiv:2408.01129*.

Qwen Team. 2025. Qwen3-next: Towards ultimate training and inference efficiency. Technical blog.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hill, and Timothy P. Lillicrap. 2020. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Ludmila Gruber, Markus Holzleitner, Milena Pavlovic, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. 2021. Hopfield networks is all you need. In *International Conference on Learning Representations*.

Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. 2025. Samba: Simple hybrid state space models for efficient unlimited context language modeling. In *International Conference on Learning Representations*.

Ricardo Buitrago Ruiz and Albert Gu. 2025. Understanding and improving length generalization in recurrent models. In *International Conference on Machine Learning*.

Yash Sarrof, Yana Veitsman, and Michael Hahn. 2024. The expressive capacity of state space models: A formal language perspective. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. 2021. Linear transformers are secretly fast weight programmers. In *Proceedings of the 38th International Conference on Machine Learning*, pages 9355–9366. PMLR.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. SCROLLS: Standardized comparison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Weigao Sun, Disen Lan, Yiran Zhong, Xiaoye Qu, and Yu Cheng. 2025a. LASP-2: Rethinking sequence parallelism for linear attention and its hybrid. *arXiv preprint arXiv:2502.07563*.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. 2025b. Learning to (Learn at test time): RNNs with expressive hidden states. In *International Conference on Machine Learning (ICML)*.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. 2023. Retentive network: A successor to Transformer for large language models. *arXiv preprint arXiv:2307.08621*.

Yutao Sun, Zhenyu Li, Yike Zhang, Tengyu Pan, Bowen Dong, Yuyi Guo, and Jianyong Wang. 2025c. Efficient attention mechanisms for large language models: A survey. *arXiv preprint arXiv:2507.19595*.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2021. Long range arena: A benchmark for efficient transformers. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, Garvit Kulshreshtha, Vartika Singh, Jared Casper, Jan Kautz, Mohammad Shoeybi, and Bryan Catanzaro. 2024. An empirical study of Mamba-based language models. *arXiv preprint arXiv:2406.07887*.

Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. 2024. LOOK-M: Look-once optimization in KV cache for efficient multimodal long-context inference. In *Findings of the Association for Computational Linguistics: EMNLP 2024*.

Junxiong Wang, Daniele Paliotta, Avner May, Alexander M. Rush, and Tri Dao. 2024a. The mamba in the llama: Distilling and accelerating hybrid models. *arXiv preprint arXiv:2408.15232*.

Penghao Wang, Yuhao Zhou, Mengxuan Wu, Panpan Zhang, Zhangyang Wang, and Kai Wang. 2025a. Data efficient any Transformer-to-Mamba distillation via attention bridge. *arXiv preprint arXiv:2510.19266*.

Shenran Wang, Timothy Tin-Long Tse, and Jian Zhu. 2025b. Understanding in-context learning beyond transformers: An investigation of state space and hybrid architectures. *arXiv preprint arXiv:2510.23006*.

Xindi Wang, Mahsa Salmani, Parsa Omidi, Xiangyu Ren, Mehdi Rezagholizadeh, and Armaghan Eshaghi. 2024b. Beyond the limits: A survey of techniques to extend the context length in large language models. *arXiv preprint arXiv:2402.02244*.

Kaiyue Wen, Xingyu Dang, and Kaifeng Lyu. 2025. RNNs are not Transformers (yet): The key bottleneck on in-context retrieval. In *International Conference on Learning Representations*.

Bingheng Wu, Jingze Shi, Yifan Wu, Nan Tang, and Yuyu Luo. 2025a. TransXSSM: A hybrid Transformer-State space model with unified rotary position embedding. *arXiv preprint arXiv:2506.09507*.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. 2025b. Retrieval head mechanistically explains long-context factuality. In *International Conference on Learning Representations*.

Yuhuai Wu, Markus N. Rabe, DeLesley Hutchins, and Christian Szegedy. 2022. Memorizing transformers. In *International Conference on Learning Representations*.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2024a. InfLLM: Unveiling the intrinsic capacity of LLMs for understanding extremely long sequences with training-free memory. In *Advances in Neural Information Processing Systems*.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. 2025. DuoAttention: Efficient long-context LLM inference with retrieval and streaming heads. In *International Conference on Learning Representations*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024b. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*.

Rui Xu, Shu Yang, Yihui Wang, Yu Cai, Bo Du, and Hao Chen. 2024. Visual Mamba: A survey and new outlooks. *arXiv preprint arXiv:2404.18861*.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. 2025. Gated delta networks: Improving mamba2 with delta rule. *arXiv preprint arXiv:2412.06464*.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. 2024. Gated linear attention transformers with hardware- efficient training. *arXiv preprint arXiv:2312.06635*.

Howard Yen and 1 others. 2024. HELMET: How to evaluate long-context language models effectively and thoroughly. *arXiv preprint arXiv:2410.02694*.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Y. X. Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. 2025. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Annual Meeting of the Association for Computational Linguistics*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297.

Zhihao Zhan, Jianan Zhao, Zhaocheng Zhu, and Jian Tang. 2025. Overcoming long-context limitations of state-space models via context-dependent sparse attention. In *Advances in Neural Information Processing Systems*.

Michael Zhang, Simran Arora, Rahul Chalamala, Alan Wu, Benjamin Spector, Aaryan Singhal, Krithik Ramesh, and Christopher Ré. 2025. LoLCATs: On low-rank linearizing of large language models. In *International Conference on Learning Representations*.

Michael Zhang, Kush Bhatia, Hermann Kumbong, and Christopher Ré. 2024a. The hedgehog & the porcupine: Expressive linear attentions with softmax mimicry. In *International Conference on Learning Representations*.

Xinrong Zhang and 1 others. 2024b. Extending long context evaluation beyond 100k tokens. *arXiv preprint arXiv:2402.13718*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. $H_2O$: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*.

Tianyu Zhao and Llion Jones. 2026. Fast-weight product key memory. *arXiv preprint arXiv:2601.00671*.

## Appendix Roadmap

The appendix answers eight reader-facing questions about evidence, scope, and differentiation; it is not an independent bibliography. Appendix A records the survey protocol and coding schema; Appendix B summarizes how cited evidence is used; Appendix C explains resource/role boundary cases; Appendix D separates capability evidence by pressure, evidence type, and caveat; Appendix E groups architectures by allocation pattern; Appendix F compares conversion and alignment targets; Appendix G records deployment budget constraints; and Appendix H differentiates this survey from related surveys. The tables are intended as claim-support infrastructure: they record what role a paper plays in the survey and which main-text claim it supports.

## A Survey Protocol and Coding Schema

This appendix makes the targeted survey protocol auditable. It is not a PRISMA-style systematic review; rather, it records how papers were selected and how each cited work was coded for the SEA analysis. Table 5 summarizes the search scope, expansion procedure, inclusion rule, and evidence-status coding used throughout the survey.

## B Evidence Use Table

This appendix answers a narrower evidence-use question: what role does each paper cluster play in the survey, and what caveat limits that use? It is organized by resource/role assignment and allocation granularity within the multi-axis framework, so each row states how a paper cluster is used rather than merely listing related work. Table 6 maps each evidence theme to its representative papers, survey use, and scope caveat.

## C Extended Resource-Role Taxonomy

This appendix answers the taxonomy-boundary question: how do specific mechanisms fit under the resource-budget and functional-role hierarchy introduced in Section 3? It distinguishes the three budgets—*active latent state*, *addressable evidence*, and *access/update bandwidth*—then describes the roles placed over them: direct lookup of retained evidence, compressed state evolution, online-updated associative prediction, external lookup, and budget/control policy. For each family it asks what resource and role are realized, at what allocation granularity, and what retrieval-relevant behavior is preserved or lost. The main text in Section 3 should be read first; this appendix is a reference layer for readers who want more detail on how specific models fit into the taxonomy.

### C.1 Explicit Content-Addressable Retrieval Memory

The defining property of this class is that retrieval is *content-addressed*: a query vector identifies relevant past positions by similarity, independent of their sequence position. The canonical instance is Transformer softmax attention (Vaswani et al., 2017), which scores all $n$ past keys against each query and aggregates the corresponding values. The KV cache stores all past key-value pairs; memory grows linearly with context.

Variants within this class include:

- **Sparse/local attention**: restrict the set of positions each query may attend to, reducing bandwidth from $O(n)$ to $O(k)$ or $O(w)$ per query. Direct content-addressable access is retained for attended positions; coverage is traded for efficiency. Representative designs include Sparse Transformers with fixed strided/block patterns (Child et al., 2019), Longformer's local+global attention mixture (Beltagy et al., 2020), and BigBird's sparse+random+global combination (Zaheer et al., 2020). All retain content-addressable access within their attended set; the allocation decision is which positions enter the attended set.
- **Sliding-window attention**: local attention with a fixed window; used in Griffin/RecurrentGemma alongside global recurrence.
- **Shared attention modules**: a single attention module applied at multiple depths (Zamba); centralized retrieval with reduced per-layer KV cost.

### C.2 Compressed-State Recurrent Memory

This class replaces the per-token KV store with a fixed-size state $h \in \mathbb{R}^d$ that is updated at each step. Arbitrary-token lookup over the full prefix is not generally supported; the guarantee is efficient inference with $O(1)$ per-step state memory.

Historical lineage:

- Transformer-XL (Dai et al., 2019): introduced segment-level recurrence by caching

Table 5: Survey protocol and coding schema.

| Protocol field | Specification |
|---|---|
| Search scope | ACL Anthology, arXiv, OpenReview, Semantic Scholar, and major conference/system proceedings, focused on 2017–2026 language-model memory work. |
| Seed queries | Hybrid language model; SSM/Mamba; linear/recurrent language model; long context; memory compression; retrieval head; KV cache; external memory; Transformer-to-SSM/Mamba distillation. |
| Expansion | Snowballing from related surveys, benchmark papers, and system papers that clarify retained evidence, compressed state, online update, external lookup, or cache/access policy. |
| Inclusion rule | Include work informing at least one SEA dimension for autoregressive text modeling: resource role, mechanism, allocation granularity, intervention phase, memory function, or mismatch risk. |
| Exclusion rule | Exclude mainly vision, genomics, time-series, graph, or broad multimodal work unless it defines a boundary case for language-model memory allocation. |
| Coding schema | Resource role; mechanism; allocation granularity; intervention phase; memory function; mismatch risk; benchmark/probe type; evidence status. |
| Evidence status | Peer-reviewed paper; preprint; technical report/blog; theory; benchmark; mechanistic analysis. Strong claims in the main text rely on peer-reviewed, formal, benchmark, or convergent evidence rather than a single recent report. |

and reusing hidden states from the previous segment; a hybrid design where the recurrent state is a compressed segment-level summary, not a per-token KV.

- HiPPO (Gu et al., 2020): optimal polynomial projection for compressing history into recurrent coefficients.
- S4 (Gu et al., 2022): structured parameterization of the state transition via diagonal-plus-low-rank matrices.
- Mamba (Gu and Dao, 2023): selective state transitions that gate how much each input token is absorbed into state.
- Mamba-2 / SSD (Dao and Gu, 2024): reformulation as structured matrix products; connects to linear attention.
- RWKV (Peng et al., 2023) and RetNet (Sun et al., 2023): alternative parameterizations using time-mixing and retention.
- Eagle/Finch (Peng et al., 2024): RWKV variants with matrix-valued states and dynamic recurrence.
- xLSTM (Beck et al., 2024): matrix-valued cell state extending LSTM to higher-dimensional compressed memory.

A recurring challenge for this class is length generalization: models trained on shorter sequences often fail to maintain coherent recurrent state at longer test-time horizons. Ruiz and Gu (2025) analyze the sources of this failure and propose strategies that improve length generalization in recurrent models while preserving their $O(1)$-memory streaming advantage.

### C.3 Online-Updated Active-State Memory

This class is a structured active-state role: it generalizes compressed-state models by allowing the recurrent state to be selectively *rewritten* rather than only accumulated. The state is typically a matrix-valued accumulator $S \in \mathbb{R}^{d \times d}$, and updates are governed by a learned rule (delta rule, gradient step, or gated difference).

Historical lineage:

- Linear attention (Katharopoulos et al., 2020): decomposes softmax attention into a matrix outer-product accumulation; equivalent to a recurrent matrix state. Schlag et al. (2021) formalize this as *fast weight programming*: linear layers are "programmed" by an outer-product write rule, making the connection to associative memory explicit.
- Induction heads (Olsson et al., 2022): mechanistic analysis of attention heads that implement "copy one token after a matching prefix"; illustrates that specific attention heads carry retrieval function, motivating the mismatch problem in Section 6.
- H3 (Fu et al., 2023): combines a shift SSM (for local dependencies) with a diagonal SSM (for long-range dependencies) plus a single

attention layer; precursor to modern hybrid designs at the layer level.

- Gated Linear Attention (GLA) (Yang et al., 2024): adds data-dependent gating for structured forgetting.
- Based (Arora et al., 2024b): simplified linear attention with explicit recall/efficiency tradeoff.
- DeltaNet / Gated DeltaNet (Yang et al., 2025): delta-rule updates; selectively rewrite associations in the matrix state.
- Test-Time Training (TTT) (Sun et al., 2025b): treats the recurrent state as the parameters of a small online-trained model; K-V pairs are "memorized" via gradient updates.
- Fast-weight product-key memory (Zhao and Jones, 2026): sparse episodic fast-weight updates; bridges online-updated recurrent memory and external memory.

The key distinction from simpler compressed-state models is that online-updated models can *selectively retain* specific associations, making them more retrieval-relevant—but they remain capacity-limited by the same active-state budget.

### C.4 External and Persistent Memory

External memory externalizes addressable evidence beyond model parameters and activation state, shifting the primary constraint from active-state capacity to storage, retrieval, and integration bandwidth. The retrieval cost shifts from compute-at-attention-time to I/O latency and index infrastructure. External retrieval systems exemplify this pattern for language models. Hybrid Associative Memories (Lufkin et al., 2026) explore a selective externalization strategy: hard-to-predict tokens are added to a KV cache, while the rest is handled by a recurrent compressive base. Engram (Cheng et al., 2026) is a boundary case for this survey: it adds conditional static lookup memory for language models, showing that the state/evidence/access framework extends beyond SSM-attention hybrids without changing the survey's central taxonomy.

### C.5 Boundary Cases and Design Tradeoffs

Several models occupy positions at the boundary between mechanism classes:

- **DuoAttention** (Xiao et al., 2025): partitions attention heads into retrieval heads (full KV) and streaming heads (compressed KV); a head-aware cache-allocation boundary case *within* the attention class.
- **RWKV-7** (Peng et al., 2025) and Eagle/Finch (Peng et al., 2024): expressive dynamic state evolution and matrix-valued recurrent states; approach online-updated recurrent memory behavior while maintaining efficient recurrent inference.
- **Bottlenecked Transformers** (Oomerjee et al., 2026): periodic KV abstraction; attention with periodic compression; boundary between full attention and compressed-state.
- **Fundamental limits** (Alman and Yu, 2025): formal lower bounds on the capabilities of subquadratic alternatives to Transformers under specified computational models; motivates caution when replacing attention-like access with cheaper sequence mechanisms.
- **Expressive limits of RNN families** (Sarrof et al., 2024): characterizes the expressiveness gap across recurrent architectures, showing how hidden-state dimensionality and activation choices bound the set of computable functions; constrains which resource/role assignments can implement retained-evidence access versus approximate retrieval.
- **Theoretical expressiveness of SSMs** (Cirone et al., 2024): formal analysis of what state-space models can and cannot compute as sequence-to-sequence functions; situates SSMs within the compressed-state class and establishes boundaries on their retrieval capacity.
- **Modern Hopfield Networks** (Ramsauer et al., 2021): reformulates softmax attention as a Hopfield energy update, explicitly connecting attention to associative memory retrieval; the memory capacity scales exponentially with the state dimension, connecting K-V lookup over retained evidence with associative active-state roles.
- **LIGER** (Lan et al., 2025): post-training linearization of a large language model into a gated recurrent structure; provides evidence

that replacement choices are layer-sensitive rather than uniformly safe.

# D Capability and Benchmark Evidence Map

This appendix answers the capability-evidence question: which pressure does each benchmark or formal result support, what type of evidence is it, and what caveat limits the claim? For each capability claim in the main text, the tables separate supporting paper(s), evidence type, scope of the claim, and caveats. Recent preprints are used as representative evidence, not as sole support for strong quantitative claims.

## D.1 Context-Access Capability Evidence

This subsection collects evidence for the most attention-sensitive capability pressure: direct access to specific past tokens, facts, or patterns. The rows combine formal copying and retrieval limits (Jelassi et al., 2024; Wen et al., 2025), synthetic recall tests (Arora et al., 2024a,c), mechanistic evidence for retrieval heads (Olsson et al., 2022; Bick et al., 2025b), and long-context factual lookup stress tests (Mitra et al., 2025). We use these results to support qualitative resource-role claims rather than single-number rankings across model families. Table 7 lists the specific evidence items and caveats behind this context-access claim.

## D.2 State-Evolution Capability Evidence

State-evolution evidence targets a different pressure: maintaining and updating compressed latent state across long sequences. The relevant failures are not necessarily missed facts, but unstable transition rules, entity-state errors, or limits of particular recurrent parameterizations. We therefore separate formal state-tracking results (Merrill et al., 2024) from empirical and theoretical evidence about transition structure, such as negative eigenvalue dynamics (Grazzi et al., 2025). Table 8 summarizes the state-evolution evidence used to distinguish update-rule limits from retrieval failure.

## D.3 Benchmark Protocol Notes

Benchmark rows are included because apparent architecture gaps can be amplified or hidden by protocol choices. Synthetic recall probes such as MQAR and needle-style tasks isolate addressable-evidence access, whereas LongBench, SCROLLS, RULER, and InfiniteBench-style suites mix retrieval, reasoning, summarization, and length extrapolation (Arora et al., 2024a; Bai et al., 2023; Shaham et al., 2022; Hsieh et al., 2024; Zhang et al., 2024b). The protocol notes indicate which evidence is diagnostic for the SEA framework and which evidence should be treated as broad end-to-end stress testing. Table 9 records these benchmark roles and caveats.

**Takeaway.** Together these rows separate context access, state evolution, and benchmark protocol effects. The evidence supports the framework by showing that failures are not a single "long-context" deficit: they depend on whether the task stresses addressable evidence, active-state updates, or evaluation protocol design.

# E Hybrid Architecture Catalogue

This appendix answers the architecture-comparison question: which allocation pattern does each hybrid system instantiate, and what mismatch risk follows from that pattern? Rows group systems by granularity, resource/role assignment, and mismatch risk rather than by model family alone. Table 10 gives the corresponding architecture catalogue.

**Takeaway.** The catalogue preserves architecture breadth while making the allocation pattern the comparison unit. The main value of the table is the mismatch-risk column: similar resource savings can fail for different reasons depending on whether allocation happens at block, layer, head, routing, or cache-policy granularity.

# F Distillation, Conversion, and Alignment Catalogue

This appendix answers the alignment question: when a Transformer is converted or distilled into a recurrent, linear-attention, or hybrid student, which behavior is actually preserved? It compares output-aligned conversion, hidden/attention alignment, retrieval-aware alignment, and scalable conversion through the lens of resource/role preservation, especially direct access to retained evidence via G&A-type retrieval behavior. For each method the table records the alignment target (logit KD, hidden-state, attention-map, retrieval-behavior, or memory-dynamics), what is concretely preserved, and whether the method addresses the context-access gap or only the output gap. The rows below emphasize methods that directly affect the main-text argument. Table 11 summarizes these conversion and alignment methods.

**Takeaway.** The table distinguishes scalable conversion from function-preserving alignment. Output or hidden-state matching can transfer broad capability, but the framework predicts that retrieval-aware targets are needed when the resource/role to preserve is direct context access.

**Design lesson.** The trend across these methods is from *output-aligned conversion* (logit KD and hidden-state alignment: MOHAWK, Llamba, Mamba in Llama), through architecture-priming conversion (Priming), toward *function-aligned conversion* (retrieval behavior: RAD; hybrid linear attention (Chen et al., 2026)). The evidence that logit alignment alone may not preserve context-access capabilities motivates retrieval-aware objectives as the next step in distillation design.

## G Systems and Deployment Evidence

This appendix answers the systems question: which deployment constraint sets the effective budget for memory allocation? Each entry is framed around the *budget constraint* it imposes or relaxes, treating deployment requirements as the source of the allocation budget that drives architectural choice. For each method the table records: the memory bottleneck targeted (KV-cache capacity, KV-cache bandwidth, recurrent-state memory, distributed-training communication, or system-level routing cost); the budget mechanism (what is evicted, compressed, or offloaded); and whether the design interacts with the main-text allocation granularities (head-level, layer-level, or system/cache policy). Methods that reduce generic FLOPs without targeting a specific memory budget are noted but deprioritized; the focus is on methods that establish a concrete cost model for hybrid architecture design. Table 12 records the deployment evidence and its caveats.

**Takeaway.** Systems methods define the budgets within which architecture choices operate. The table separates methods that make attention cheaper from methods that change the state/evidence split, preventing kernel or cache advances from being mistaken for architectural allocation.

**Key design lesson.** Kernel-level and cache-management methods are *complementary* to architectural hybridization: FlashAttention improves the per-step cost of attention but does not remove KV-cache growth; hybrid architectures remove most KV-cache growth but benefit additionally from FlashAttention within their attention layers. The two strategies should be combined, not treated as alternatives.

## H Related Surveys and Differentiation

This appendix answers the differentiation question: what does this survey add beyond existing SSM/Mamba, long-context, efficient-attention, and augmented-LM surveys? The organizing claim: existing surveys primarily catalogue architectures, applications, or efficiency techniques; this survey provides a memory-allocation *framework* that explains *why* hybridization arises and *how* design choices allocate active state, addressable evidence, and access/update bandwidth to retrieval, update, and policy roles. Table 13 summarizes the comparison against neighboring survey lines.

**Takeaway.** The comparison shows that prior surveys cover important neighboring literatures, but usually organize them by architecture family, efficiency method, or context-extension technique. The present survey's added value is the resource-budget and functional-role explanation of why hybridization arises.

**Novelty statement.** This survey's contribution is not coverage breadth alone. It is the *function-preserving memory-allocation framework*: the unified resource-budget and functional-role framework (Section 3), the capability-pressure analysis (Section 4), and the multi-axis design map (Section 5). Together, these three elements explain why hybridization often arises as a resource-allocation response, which existing SSM/Mamba surveys do not attempt.

Table 6: Evidence roles and caveats used in the survey.

| Evidence theme | Representative papers | Use in survey | Scope / caveat |
|---|---|---|---|
| Mechanistic retrieval gap | G&A and retrieval-head analyses (Bick et al., 2025b; Wu et al., 2025b; Michalak and Abreu, 2025); mechanistic evaluations (Arora et al., 2025; Wang et al., 2025b) | Supports Sections 4 and 6: context access depends on retrieval-specialized attention components, so coarse hybridization can preserve perplexity while losing retrieval function. | Recent mechanistic evidence; use strongest claims for qualitative mismatch and head-function arguments rather than exact universal laws. |
| Theoretical unity and capability separation | SSD / structured-matrix view (Dao and Gu, 2024); RNN–Transformer separation (Wen et al., 2025); copying and subquadratic limits (Jelassi et al., 2024; Alman and Yu, 2025) | Grounds Section 3: attention and SSMs can share algebraic structure while differing in addressable evidence, latent-state capacity, and access bandwidth. | Formal models differ from deployed LLMs; claims are used to motivate capability pressure, not to declare blanket impossibility. |
| Compressed-state sequence modeling | S4/HiPPO/Mamba line (Gu et al., 2022; Gu and Dao, 2023; Gu et al., 2020); recurrent baselines (Peng et al., 2023; Sun et al., 2023; Beck et al., 2024; Qin et al., 2024; Peng et al., 2025) | Defines compressed relational/dynamical state as the active latent-state budget's main functional role and supplies the efficiency anchor for hybrid architectures. | Strong for streaming and compression; direct content-addressable access remains task-dependent and often requires additional evidence. |
| Online-updated active-state role | Linear attention and gated updates (Katharopoulos et al., 2020; Yang et al., 2024, 2025); TTT / fast-weight variants (Sun et al., 2025b; Zhang et al., 2024a; Zhao and Jones, 2026) | Supports the online-updated predictor / associative-map lens as a structured active-state role in Section 3.3 and the K-V allocation open problem. | These methods store learned statistics or predictor parameters, not a full token-level KV table; retrieval fidelity depends on update capacity. |
| Capability-pressure evidence | MQAR and recall benchmarks (Arora et al., 2024a); Mamba ICL studies (Grazzi et al., 2024; Park et al., 2024); state-tracking work (Merrill et al., 2024; Grazzi et al., 2025; Ruiz and Gu, 2025) | Supports Section 4: context access, smoother ICL regimes, and state evolution expose different resource/role requirements. | Evidence is heterogeneous across theory, synthetic tasks, and empirical benchmarks; avoid hard checkmark-style claims. |
| Layer/block hybrid allocation | Interleaving and production hybrids (Lieber et al., 2024; Jamba Team, 2024; Ren et al., 2025; NVIDIA, 2025; Gu et al., 2025; MiniMax Team, 2025; Qwen Team, 2025; MiniCPM Team, 2026); long-context upcycling (Fashi et al., 2026); local-global hybrids (De et al., 2024; Botev et al., 2024) | Supports the Section 5 claim that current systems mostly allocate direct access to retained evidence at layer or block granularity. | Attention ratio and placement are not equivalent to retrieval-function preservation; detailed per-system comparison is in Appendix E. |
| Head/shared/routing allocation | Head-level designs (Dong et al., 2025; Falcon LLM Team, 2025); shared or amortized attention examples (Glorioso et al., 2024); routing / analysis work (Li et al., 2025b; Wu et al., 2025a; Zhan et al., 2025; Ling Team, 2025; Waleffe et al., 2024) | Shows that allocation can occur within layers, through reusable or amortized attention mechanisms, or through input-dependent routing rather than only periodic layer interleaving. | Current routing and sharing evidence rarely supervises future retrieval relevance directly; mismatch risk remains central. |
| Training-time alignment and conversion | Conversion and distillation methods (Bick et al., 2024, 2026; Goldstein et al., 2025; Bick et al., 2025a; Wang et al., 2024a; Zhang et al., 2025; Nguyen et al., 2025; Chen et al., 2026; Wang et al., 2025a; Lan et al., 2025; Chattopadhyay et al., 2026) | Supports Section 6: replacing attention requires deciding which teacher behavior, heads, states, or retrieval traces to preserve. | Output-level alignment may not preserve context-access behavior; retrieval-aware objectives are promising but recent. |
| Systems and cache-budget evidence | Exact-attention kernels and sparse/cache policies (Dao et al., 2022; Dao, 2024; Sun et al., 2025a; Xiao et al., 2024b,a, 2025; Wan et al., 2024; Deng et al., 2024; Yuan et al., 2025; Nawrot et al., 2025; Mitra et al., 2025; Cabannes et al., 2025; Li et al., 2024b) | Supports Section 5: deployment creates concrete budgets for KV capacity, access bandwidth, recurrent-state size, and cache policy. | Kernel improvements and cache policies complement architecture; they do not by themselves solve resource-role allocation. |
| External and emerging memory regimes | External / hybrid associative memory candidates (Zhao and Jones, 2026; Lufkin et al., 2026; Dubinin et al., 2026; Behrouz et al., 2026) | Supports open problems around sparse retained evidence, growing recurrent memory, and hybrid streaming associative memory. | Recent or speculative evidence; treated as future design space rather than solved K-V allocation. |

Table 7: Context-access capability evidence.

| Evidence item | Source / type | Claim supported | Caveat |
|---|---|---|---|
| Exact copying | Jelassi et al. 2024 (Jelassi et al., 2024)<br>Theory + empirical | Fixed-state models cannot copy from arbitrary positions as $n \to \infty$. | Formal model assumes unbounded $n$; practical impact varies with context length. |
| Associative recall (MQAR) | Arora et al. 2024 (Arora et al., 2024a)<br>Empirical benchmark | SSMs and linear attention underperform attention on MQAR across model sizes. | Performance gap narrows with larger state; avoid relying on exact numeric deltas. |
| In-context retrieval (general) | Wen et al. 2025 (Wen et al., 2025)<br>Theory + empirical | Linear RNNs face an exponential memory bottleneck for in-context retrieval. | Claim is scoped to linear RNNs; selective SSMs such as Mamba partially mitigate it. |
| ICL with smooth targets | Grazzi et al. 2024 (Grazzi et al., 2024)<br>Empirical | Mamba performs ICL for low-dimensional smooth function classes. | Does not imply discrete content-addressable lookup or arbitrary label mapping. |
| Retrieval-heavy ICL | Bick et al. 2025 (Bick et al., 2025b)<br>Mechanistic + empirical | G&A heads appear critical for retrieval-heavy ICL and are absent in pure SSMs. | Boundary cases such as TTT or DeltaNet require caution. |
| Long-context factual lookup | Hsieh et al. 2025 (Mitra et al., 2025)<br>Long-context benchmarks | SSM/hybrid models show factual-retrieval drops at very long contexts. | Exact deltas are benchmark-dependent; needle-style tasks may favor attention. |
| Recall gap closure (JRT-RNN) | Arora et al. 2024 (Arora et al., 2024c)<br>Empirical | Just Read Twice substantially narrows recurrent-model recall gaps by adding a second sequential pass. | Holds for specific task formats; general lookup still needs attention or external KV. |
| Induction head mechanism | Olsson et al. 2022 (Olsson et al., 2022)<br>Mechanistic analysis | Induction heads are dedicated attention circuits for in-context pattern copying. | Findings are for Transformer attention; analogous hybrid circuits remain an open question. |

Table 8: State-evolution capability evidence.

| Evidence item | Source / type | Claim supported | Caveat |
|---|---|---|---|
| Entity tracking | Merrill et al. 2024 (Merrill et al., 2024)<br>Formal-language theory | Common SSM parameterizations can share bounded-depth Transformer limits for state tracking. | Idealized computational model; not a blanket claim about arbitrary gated RNNs. |
| Algorithmic state transitions | Merrill et al. 2024 (Merrill et al., 2024)<br>Theoretical | Common SSMs or constrained state-space updates may fail on compositional transition sequences. | Scope is update-rule constraints, not arbitrary nonlinear recurrent computation. |
| Negative-eigenvalue state tracking | Grazzi et al. 2025 (Grazzi et al., 2025)<br>Empirical + theoretical | Negative eigenvalues in linear RNNs expand state-tracking capability. | State tracking depends on transition structure, not recurrence alone. |

Table 9: Benchmark protocol notes.

| Benchmark | Paper | What it measures | Caveat |
|---|---|---|---|
| MQAR (Multi-Query Assoc. Recall) | Arora et al. 2024 (Zoology) (Arora et al., 2024a) | Associative recall fidelity across context lengths | Isolates discrete recall; does not measure broader language quality |
| Recall benchmark (Zoology suite) | Arora et al. 2024 (Arora et al., 2024a) | Recall precision for SSM vs. attention families | Useful for mechanism stress tests; exact deltas are setup-dependent |
| LongBench | Bai et al. 2023 (Bai et al., 2023) | Multi-task long-context comprehension (bilingual) | End-to-end benchmark; task mixture can obscure specific retrieval failures |
| LRA | Tay et al. 2021 (Tay et al., 2021) | Efficient-sequence-model stress tests over long inputs | Useful historical benchmark; less aligned with modern generative LMs |
| SCROLLS | Shaham et al. 2022 (Shaham et al., 2022) | Long-document summarization, QA, and NLI-style tasks | Task realism improves, but memory function is less isolated |
| Needle-in-a-haystack | Various; Hsieh et al. 2025 (Mitra et al., 2025) | Long-context factual retrieval (synthetic) | Standard protocol; numbers should be treated cautiously |
| RULER | Hsieh et al. 2024 (Hsieh et al., 2024) | Parametric evaluation of "real" context window size under retrieval pressure | Controlled benchmark; hybrid vs. pure-attention gaps measurable |
| L-Eval / InfiniteBench / HELMET | Chen et al.; Zhang et al.; Yen et al. (Chen et al., 2023; Zhang et al., 2024b; Yen et al., 2024) | Long-context application suites and 100K+ context stress tests | Broader coverage; still needs language/domain/function tagging |

Table 10: Hybrid architecture catalogue by allocation pattern.

| Allocation pattern | Systems | Resource/role assignment | Mismatch risk / caveat |
|---|---|---|---|
| Block-local retained-evidence access plus global recurrence | Griffin (De et al., 2024); RecurrentGemma (Botev et al., 2024) | Local-window KV provides direct access to recent K-V evidence while gated recurrence compresses the global tail. | Attractive for streaming and state evolution, but far-past retrieval depends on compressed state quality. |
| Periodic layer interleaving | Jamba, Jamba 1.5 (Lieber et al., 2024; Jamba Team, 2024); Samba (Ren et al., 2025); Nemotron-H, Jet-Nemotron (NVIDIA, 2025; Gu et al., 2025); MiniMax-01 (MiniMax Team, 2025); Qwen3-Next (Qwen Team, 2025); MiniCPM-SALA (MiniCPM Team, 2026) | Full, sparse, or gated attention/linear-attention layers act as retrieval anchors inside mostly recurrent or efficient-mixer stacks. | Attention ratio is not enough: task-agnostic placement can miss retrieval-critical components (Bick et al., 2025b; Ling Team, 2025); Qwen3-Next should be read as technical-blog evidence. |
| Shared or amortized attention | Zamba (Glorioso et al., 2024) | Reusable or amortized attention mechanisms reduce per-layer attention cost while preserving some direct retained-evidence access. | Sharing/amortization may under-cover layer- or token-specific retrieval, so coverage should be evaluated under harder recall tasks. |
| Head-level fusion | Hymba (Dong et al., 2025); Falcon-H1 (Falcon LLM Team, 2025) | Softmax heads retain K-V evidence while recurrent or SSM heads carry compressed dynamics within a layer. | The split helps only if attention heads align with retrieval-critical functions; current designs rarely supervise this alignment directly. |
| Dynamic routing | TransMamba (Li et al., 2025b); TransXSSM (Wu et al., 2025a); HAX/CDSA (Zhan et al., 2025) | Input-dependent routing chooses between attention-like and recurrent computation or selects query-conditioned sparse evidence. | Most directly resembles learned allocation, but routing stability and future-relevance supervision remain open. |
| Long-context upcycling | HyLo (Fashi et al., 2026); MiniCPM-SALA (MiniCPM Team, 2026) | Converts or continues pretrained Transformer models into hybrids with sparse/efficient attention and linear or SSM-style blocks. | Recent preprint evidence; useful for intervention-phase analysis but still needs function-specific diagnostics. |
| K-V/sub-head-adjacent growing memory | Memory Caching (Behrouz et al., 2026); hybrid associative memory (Lufkin et al., 2026); fast-weight product-key memory (Zhao and Jones, 2026) | Cached recurrent checkpoints, selective KV growth, or sparse fast-weight access approach fine-grained memory allocation. | These are adjacent to K-V allocation but do not yet solve literal Transformer K-V-pair selection. |
| Function-aware analysis and baselines | Every Attention Matters (Ling Team, 2025); HGRN2 (Qin et al., 2024); empirical Mamba/hybrid studies (Waleffe et al., 2024) | Analysis and recurrent baselines clarify which layers or mechanisms matter when attention is reduced. | Useful as diagnostic evidence rather than a standalone allocation policy. |

Table 11: Distillation, conversion, and alignment catalogue.

| Method | Year | Alignment target | Preserved behavior | Scale/status | Caveat |
|---|---|---|---|---|---|
| MOHAWK (Bick et al., 2024) | NeurIPS 2024 | Logit KD; weight init.; SSD approx. | General capability; retrieval evidence remains preliminary | Up to 2.8B | Retrieval gap not directly measured |
| RAD (Bick et al., 2026) | 2026 preprint | Retrieval-behavior alignment | G&A-type context-access behavior | Recent preprint | Retrieval target explicit; full evaluation preliminary |
| Goldstein et al. (Goldstein et al., 2025) | COLM 2025 | Low-token-cost fine-tuning | General capability at scale | Large-scale conversion | Retrieval-specific coverage remains limited |
| Priming (Chattopadhyay et al., 2026) | 2026 preprint | Layer selection; weight transfer; layerwise alignment | Hybrid SSM conversion and SSM-layer comparison | 8B–32B; 128K context | Recent preprint; function-specific preservation remains indirect |
| Llamba (Bick et al., 2025a) | 2025 preprint | MOHAWK-style Transformer-to-Mamba distillation | General capability and efficient recurrent deployment | 3B–8B | Does not explicitly preserve attention heads or G&A behavior |
| AttBridge (Wang et al., 2025a) | 2025 preprint | Data-efficient Transformer-to-Mamba conversion | General capability; small-data conversion efficiency | Small-to-medium scale | Context-access preservation not directly targeted |
| Mamba in the Llama (Wang et al., 2024a) | 2024 preprint | Knowledge distillation (logits + hidden) | General capability | Up to 8B | No retrieval-specific alignment |
| LoLCATs (Zhang et al., 2025) | 2024 | Progressive linearization (attention $\rightarrow$ SSM) | General capability; low-cost conversion | Up to 70B | Retrieval role may be diluted by linearization |
| Lizard (Nguyen et al., 2025) | 2025 | Efficient linearization framework | Reduced calibration cost | Recent framework | Retrieval-specific evidence limited |

Table 12: Systems and deployment evidence.

| Method | Bottleneck | Mechanism | Gain | Source | Caveat |
|---|---|---|---|---|---|
| FlashAttention (Dao et al., 2022) | Attention memory bandwidth | IO-aware kernel fusion; avoids materializing attn. matrix | Kernel-level | NeurIPS 2022 | Kernel-only; preserves dense attention |
| FlashAttention-2 (Dao, 2024) | Attention memory bandwidth | Improved parallelism and tiling | Kernel-level | ICLR 2024 | Kernel-only; does not change evidence allocation |
| LASP-2 (Sun et al., 2025a) | Distributed training memory for linear attn. | Sequence parallelism with hybrid-aware communication | Kernel + distributed | 2025 preprint | Training-system support, not an allocation rule |
| StreamingLLM (Xiao et al., 2024b) | KV-cache for streaming | Attention sink + sliding window; no KV cache growth | Cache-management | 2023 | Cache policy; does not alter model architecture |
| InfLLM (Xiao et al., 2024a) | KV-cache at long context | Block-level KV memory management (training-free) | Cache-management | 2024 | Retrieval policy outside core architecture |
| DuoAttention (Xiao et al., 2025) | KV-cache (retrieval vs. streaming heads) | Full KV only for retrieval heads; compressed KV for streaming | Architectural + cache | 2024 | Depends on correct head classification |
| Look-M (Wan et al., 2024) | Multimodal KV cache | Key merging for visual tokens | Cache-management | 2024 | Multimodal-boundary evidence |
| Sparse attn. ($n^C$ theory) (Deng et al., 2024) | Attention retrieval bandwidth | Natural $n^C$ sparsity reduces attended tokens | Architectural (theory) | 2024 | Theoretical sparsity model; use qualitatively |
| EAGLE (Li et al., 2024b) | Autoregressive decoding latency | Speculative decoding via a lightweight draft model; accepts or rejects proposed tokens using the target model's KV cache | Decoding throughput ($3\times$+) | ICML 2024 | Decoding-only speedup; not memory allocation |
| Hybrid architectures | KV-cache at long context | Shift majority of computation to $O(1)$ recurrent state | Architectural | Jamba, Griffin, Nemotron-H | Architecture-level budget tradeoff |

Table 13: Related surveys and differentiation.

| Survey | Year | Organizing principle | This survey's difference |
|---|---|---|---|
| "A Survey of Mamba" (Qu et al.) (Qu et al., 2024) | 2024 | SSM/Mamba architecture evolution and applications | We do not organize by SSM family; our axes are resource budget and allocation pattern |
| "A Comprehensive Survey on Long Context Language Modeling" (Liu et al.) (Liu et al., 2025) | 2025 | Long-context techniques covering attention, compression, and retrieval | We go beyond efficiency to explain resource-role gaps that motivate hybridization |
| "Visual Mamba: A Survey and New Outlooks" (Xu et al.) (Xu et al., 2024) | 2024 | Mamba applications in computer vision | Vision hybrids are noted but not surveyed; our focus is language-model retrieval |
| "Efficient Attention Mechanisms for Large Language Models: A Survey" (Sun et al.) (Sun et al., 2025c) | 2025 | Linear and sparse attention mechanisms, including hardware-aware efficient attention | We treat efficient attention as one memory-allocation substrate, not the organizing axis |
| "Augmented Language Models: a Survey" (Mialon et al.) (Mialon et al., 2023) | 2023 | External tools, reasoning modules, and non-parametric augmentation | We include externalized evidence as an addressable-evidence regime, while keeping architecture-level memory allocation central |
| "A Survey on Large Language Model Acceleration based on KV Cache Management" (Li et al.) (Li et al., 2025a) | 2025 | Token-, model-, and system-level KV-cache management | We use cache management as the system-policy end of the same allocation taxonomy |
| "Beyond the Limits" (Wang et al.) (Wang et al., 2024b) | 2024 | Techniques for extending LLM context length | We connect long-context methods to resource/role requirements rather than cataloguing extension techniques |
| "The End of Transformers? On Challenging Attention and the Rise of Sub-Quadratic Architectures" (Fichtl et al., 2025) | 2025 | Efficiency-oriented comparison of sub-quadratic models | Our framework is capability-centric, not efficiency-centric; we explain the behavioral differences |