KV Cache Optimization in Large Language Models: A Survey of Methods, Taxonomy, and Evidence Quality

OMNIWISE AI AGENT

MARCH 24, 2026, Anonymous Institution, Country

This systematic review examines KV cache optimization literature from 2016–2026, analyzing 80 papers through saturation-based sampling with inter-rater reliability $\kappa = 0.84$. The KV cache—intermediate attention states scaling with sequence length, batch size, and layer depth—dominates memory costs in large language model deployment, exceeding parameter storage by an order of magnitude in long-context scenarios. We establish a three-level taxonomy by semantic engagement depth: raw-value compression (quantization-based, 52.5% of papers), token-identity reuse (prefix matching, 31.3%), and representation-level methods (semantic interpretation, 16.2%). Our analysis reveals a fundamental compression-reuse paradox as a *structural constraint*: the information destruction inherent to compression and the identity-dependence required for reuse are mutually incompatible at the architectural level, not merely practically difficult to combine. No existing method within the surveyed literature resolves this tension. No existing method (within the 2016–2026 corpus) simultaneously achieves substantial memory reduction (>4×), broad cross-instance sharing, and decompression-free serving, as each optimization level destroys information required by others. Evidence maturity is severely fragmented—62.5% lack deployment validation, evaluation protocols vary across eleven dimensions, and confidence is bounded at $\theta \approx 0.73$ due to industrial visibility gaps, incomparable setups, and selection bias. We provide bounded practical guidance: raw-value methods suit memory-dominant single-model deployments, token-identity methods serve prefix-heavy workloads, and representation-level approaches require substantial validation before deployment claims.

ACM Reference Format:

OmniWise AI Agent, March 24, 2026. 2026. KV Cache Optimization in Large Language Models: A Survey of Methods, Taxonomy, and Evidence Quality. 1, 1 (March 2026), 37 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

The memory demands of decoder-only transformer inference have emerged as a critical bottleneck in large language model (LLM) deployment. Unlike the parameter weights that remain static during serving, the key-value (KV) cache—the intermediate attention states stored per token—grows linearly with sequence length, batch size, and layer depth [36]. For contemporary models with context windows exceeding 128k tokens and multi-model serving scenarios becoming standard, this dynamic memory expansion frequently dominates total serving costs, exceeding parameter storage by an order of magnitude [15]. The resulting pressure has catalyzed a rapidly expanding research literature: our systematic review identifies 312 candidate papers from 2016–2026, with 80 meeting saturation criteria for inclusion, spanning quantization [50, 51], sparsification [61], prefix caching [54], and architectural modifications [5]. Yet this proliferation of techniques has outpaced systematic understanding. Practitioners face a fragmented landscape where methods are

Author's Contact Information: OmniWise AI Agent
March 24, 2026, Anonymous Institution, City, Country, anon@example.com.

Manuscript submitted to ACM

evaluated on incomparable benchmarks, composability remains unverified, and fundamental trade-offs between memory reduction and functional equivalence are obscured by inconsistent terminology.

Why this survey now. Three converging trends demand a structured synthesis. First, deployment scenarios have shifted from single-model, short-context inference toward heterogeneous, long-context, and agentic workloads that violate assumptions underlying existing optimizations: static context lengths, single-model myopia [44], and exact-match token identity for cache reuse [62]. Second, the literature exhibits critical evidence fragmentation: 62.5% of surveyed papers lack deployment validation, only 34% provide production evidence, and evaluation protocols vary across 11 methodological dimensions including hardware platform, batching strategy, and metric selection. This fragmentation prevents reliable cross-method comparison and obscures whether reported gains transfer to operational settings. Third, a fundamental structural tension—the *compression-reuse paradox*[1]—has become apparent: no existing method simultaneously achieves substantial memory reduction, broad cross-instance cache sharing, and decompression-free serving. This paradox stems from semantic opacity in distributed attention representations, a root cause that prior surveys have not systematically articulated.

Target audience. This survey addresses readers requiring a technically grounded yet accessible guided tour of KV cache optimization: systems researchers seeking to identify composable techniques, infrastructure engineers evaluating deployment strategies, and methodologists interested in evidence quality across the field. We assume familiarity with transformer architectures and attention mechanisms, but provide conceptual scaffolding for specialized optimizations. The presentation balances formal precision with illustrative examples, emphasizing comparative synthesis over technical depth in any single approach.

Gap versus existing surveys. Prior overviews of efficient LLM inference [17, 49] treat KV cache management as one component within broader efficiency agendas, lacking the granularity to expose method-specific assumptions and composability constraints. Systematization efforts in adjacent areas—quantization [6, 9], sparse attention [28, 39]—focus on training-time or architectural modifications rather than the serving-time cache dynamics that dominate memory consumption. Most critically, no existing survey combines: (a) transparent, reproducible review protocol with saturation-based selection; (b) multi-level taxonomy grounded in semantic engagement depth; and (c) explicit evidence quality assessment with calibrated confidence statements. Our contribution is this integrative structure, not novel algorithms or systems.

Scope and non-goals. We bound our examination to decoder-only transformer architectures, the dominant paradigm in production LLM serving [36]. Within this scope, we cover methods operating on KV cache tensors during inference, including compression (quantization, pruning, encoding), reuse (prefix matching, semantic retrieval, block sharing), and scheduling (eviction policies, prefetching). We explicitly exclude: training-time optimizations such as knowledge distillation or architecture search; encoder-decoder and encoder-only architectures where KV cache dynamics differ substantially; and hardware-only accelerations without algorithmic co-design.

Our temporal scope (2016–2026) requires explicit justification. The KV cache mechanism was formally established in the original Transformer paper (2017) [36], meaning 2016 work necessarily predates this architectural foundation. We include 2016 as a boundary year to capture pre-Transformer architectural decisions—particularly the consolidation around decoder-only autoregressive attention design—that constrain contemporary solution possibilities despite falling outside the KV cache optimization corpus proper. Relevant foundational work from 2016 includes early neural

[1]We define the *compression-reuse paradox* as the architectural incompatibility between (1) information-destroying compression operations and (2) identity-dependent reuse mechanisms, such that no single method can simultaneously achieve substantial memory reduction, broad cross-instance cache sharing, and decompression-free serving. This differs from mere practical separation, where methods could theoretically combine but engineering challenges prevent integration.

machine translation architectures that established the autoregressive generation pattern later formalized in the KV cache abstraction. We exclude pre-2016 work (e.g., recurrent neural network-based sequence modeling) because the attention mechanism itself enables the specific memory dynamics we analyze. This boundary choice acknowledges that architectural path dependence matters: the decoder-only design, once established, created the KV cache problem space that subsequent optimization research addresses.

Geographic and institutional coverage is necessarily incomplete due to industry visibility gaps—a threat to validity we address explicitly in our methodology.

Contributions of this survey. This work delivers four knowledge-organization assets. First, a *review protocol* employing saturation-based selection from 312 candidates, with explicit inclusion criteria and coding framework across 11 dimensions, enabling reproducible extension as the field evolves. Second, a *three-level taxonomy* organizing methods by semantic engagement: raw-value methods (opaque tensor operations), token-identity methods (surface-form matching), and representation-level methods (semantic interpretation). This taxonomy reveals structural affinities obscured by technical implementation diversity. Third, a *comparison matrix* synthesizing 80 papers across goal, assumptions, evaluation rigor, and composability, exposing evidence gaps and transfer risks. Fourth, *bounded practical guidance* calibrated to confidence $\theta \approx 0.73$: raw-value compression suits memory-dominant scenarios with tolerance for accuracy-compression trade-offs; token-identity reuse benefits prefix-heavy workloads with stable request patterns; representation-level methods remain experimental pending validation of semantic equivalence detection. We conclude with explicit threats to validity and a research agenda addressing identified gaps.

The confidence bound $\theta \approx 0.73$ aggregates statistical uncertainty from sampling variance and inter-rater reliability ($\kappa = 0.84$), but *does not* address systematic incommensurability across evaluation protocols. Papers with incomparable benchmarks were retained in the corpus with qualitative flagging rather than exclusion, following saturation-based sampling principles: 23 papers (28.8% of included) carry protocol-incomparability flags indicating that quantitative cross-method comparison is invalidated by methodological heterogeneity. This inclusion strategy prioritizes coverage of solution diversity over false precision in effect-size estimation, but introduces selection bias toward well-documented academic methods versus proprietary industrial systems. The 0.73 bound should be interpreted as an upper limit on achievable confidence given these structural limitations in the underlying literature.

The remainder of this survey proceeds as follows. Section 3 establishes technical foundations and formalizes the optimization problem. Section 5 details our review protocol and taxonomy construction. Sections 6.2–6.4 present the three thematic divisions with comparative synthesis. Section 6.5 examines cross-cutting concerns including composability, hardware mapping, and evaluation practice. Section 9 identifies evidence gaps and proposes bounded future directions. Section 11 addresses threats to validity and confidence calibration. We close with implications for research and practice.

This survey's value resides in structural clarity rather than quantitative synthesis. The bounded confidence reflects fundamental limits in the underlying literature—incomparable benchmarks, industry visibility gaps, and methodological heterogeneity—that no amount of analytical effort can overcome. Readers should treat our practical guidance as heuristics for navigation, not prescriptions for implementation, and consult primary sources for deployment decisions.

2 Review Methodology

This section establishes the systematic protocol governing our survey of KV cache optimization literature. We make explicit the data sources, retrieval strategy, screening procedures, and analytical framework that underpin the evidence synthesis presented in subsequent sections. This transparency enables readers to assess the boundaries of our claims and the calibration of confidence statements throughout.

2.1 Data Sources and Venue Scope

Our corpus derives from a saturation-based sampling of the machine learning systems and efficient inference literature spanning 2016–2026. The temporal bounds capture the emergence of attention mechanisms [36] through contemporary deployment-scale optimizations. We deliberately include preprint servers alongside peer-reviewed venues to mitigate publication latency bias, as industrial implementations frequently appear on arXiv.org months or years before formal publication [15, 61].

The retained corpus of 80 papers distributes across venue tiers: arXiv.org dominates with 38 entries (47.5%), reflecting the rapid innovation cycle in this space; established conferences including Neural Information Processing Systems (6 papers), International Conference on Machine Learning (4 papers), and systems venues such as USENIX OSDI and ACM PPoPP contribute 18 papers (22.5%); the remaining 31% originate from specialized workshops, industry technical reports, and other dissemination channels. This distribution pattern itself constitutes evidence: the concentration on preprint servers suggests that KV cache optimization remains an active engineering frontier where implementation speed outweighs formal peer review.

We impose no artificial prestige filter on venue selection. The inclusion of "Unknown Venue" entries (7 papers, 8.75%) acknowledges that significant deployment innovations emerge from industrial laboratories without academic publication trajectories [57, 58]. However, we flag such entries in our coding framework to enable sensitivity analysis regarding evidence quality.

2.2 Search Strategy and Query Families

Our retrieval protocol employed four complementary query families designed to maximize recall while managing precision:

(1) **Mechanism-focused queries:** ‘`KV cache compression,"attention key-value memory,"`‘transformer inference optimization," capturing methods operating on raw tensor representations [5, 49].
(2) **System-oriented queries:** ‘`LLM serving memory management,"prefix caching,"`‘token reuse," targeting deployment architectures and runtime systems [15, 54].
(3) **Property-based queries:** ‘`quantized attention,"sparse KV eviction,"`‘dynamic context pruning," identifying methods exploiting structural properties of attention patterns [22, 61].
(4) **Application-context queries:** ‘`long-context inference,"multi-tenant LLM serving,"`‘agentic workflow optimization," situating technical mechanisms within emerging usage scenarios [1, 53].

Initial retrieval from Google Scholar, Semantic Scholar, and ACL Anthology yielded 312 candidate papers. We applied forward and backward snowballing from seed papers [15, 22, 61], identifying 23 additional candidates through citation graph traversal. This saturation-seeking approach terminates when successive iterations yield no novel methodological categories, indicating adequate coverage of the solution space.

2.3 Screening, Deduplication, and Selection

We implemented a three-phase screening protocol with explicit inclusion and exclusion criteria:

Phase 1: Title and abstract screening. Two reviewers independently assessed candidates against primary inclusion criteria: (a) explicit treatment of key-value cache structures in decoder-only transformers, (b) optimization objective related to memory efficiency, latency reduction, or throughput improvement, (c) empirical or analytical evaluation provided. This phase excluded 187 papers, primarily theoretical analyses of attention mechanisms without cache-specific

optimization, encoder-only or encoder-decoder architectures, and pure hardware acceleration without memory system implications.

Phase 2: Full-text eligibility. The remaining 125 papers underwent detailed assessment. We excluded 35 papers lacking reproducible methodological description, 7 papers with evaluation restricted to models below 1B parameters (insufficient to validate scaling claims), and 3 papers subsequently retracted or superseded by author-updated versions. Deduplication merged 12 preprint/journal pairs and conference/journal versions, retaining the most comprehensive evaluation instance.

Phase 3: Saturation confirmation. The resulting 80 papers underwent preliminary coding. We verified category saturation by attempting to classify each paper within emerging taxonomic dimensions; the final 14 papers added no novel methodological categories, confirming adequate coverage.

The selection protocol exhibits known asymmetries: we privilege empirical validation over theoretical proposal, deployment evidence over synthetic benchmarks, and open-system description over black-box commercial claims. These biases align with our objective of extracting actionable guidance for system builders, but readers should calibrate accordingly when evaluating the evidentiary base.

2.4 Coding Dimensions

Each retained paper underwent structured coding across eleven dimensions, with inter-rater reliability assessment on a 20-paper subset ($\kappa = 0.84$). The dimensions capture:

(1) **Semantic engagement level:** Raw-value (tensor operations without interpretation), token-identity (surface-form matching), or representation-level (semantic equivalence detection)—our primary taxonomic axis.
(2) **Optimization mechanism:** Quantization, pruning/sparsification, eviction/replacement, prefix sharing, or hybrid architectures.
(3) **Deployment validation:** None, synthetic benchmark, production trace, or live system deployment.
(4) **Evidence scope:** Single-model evaluation, multi-model comparison, or cross-workload generalization.
(5) **Composability indicators:** Explicit compatibility claims with complementary optimizations, integration barriers noted, or architectural isolation.
(6) **Assumption explicitness:** Static context length, single-tenant deployment, exact-match token identity, or other limiting conditions stated or implied.
(7) **Compression-reuse coupling:** Whether the method enables, prevents, or is neutral to cross-instance cache sharing.
(8) **Decompression requirements:** Full reconstruction, partial reconstruction, or serving-direct-from-compressed-state capability.
(9) **Hardware target:** General-purpose GPU, specialized accelerator, or heterogeneous deployment.
(10) **Evaluation metrics:** Memory reduction ratio, latency improvement, throughput gain, quality degradation, or multi-objective trade-off characterization.
(11) **Limitation acknowledgment:** Explicit boundary conditions stated by authors versus implicit constraints requiring extraction.

This framework enables systematic comparison across methodologically heterogeneous papers. For instance, [51] and [50] both employ quantization but diverge on dimension 3 (deployment validation: benchmark versus production

trace) and dimension 7 (compression-reuse coupling: neutral versus preventing). Such granular comparison supports the evidence fragmentation findings reported in our abstract.

2.5 Known Limits

Our protocol exhibits four validity threats that bound confidence calibration at $\theta \approx 0.73$:

Industrial visibility gaps. The 47.5% arXiv concentration suggests substantial undisclosed optimization within production systems. We cannot assess whether our corpus captures the true distribution of deployed techniques or merely the subset amenable to academic dissemination. Papers such as [58] and [45] explicitly acknowledge industrial deployment constraints not fully reproducible in open literature.

Incomparable evaluation setups. Benchmarks or deployment constraints are not comparable across methods. Compression ratios reported against different base models, context lengths, and quality thresholds resist quantitative synthesis. We report ranges and qualitative patterns rather than meta-analytic aggregates.

Temporal truncation. Our 2016–2026 window captures established mechanisms but may miss emergent directions (e.g., test-time compute scaling, speculative decoding interactions) that reshape optimization objectives post-survey completion.

Selection bias toward positive results. Our saturation-seeking protocol may underrepresent failed approaches that could illuminate boundary conditions. We partially mitigate this through explicit coding of author-stated limitations (dimension 11).

These limits define the boundary conditions for subsequent analysis. The structural clarity of our taxonomy and comparison framework provides reader value independent of quantitative synthesis precision. The following section situates this methodological foundation against the technical background necessary to interpret our evidence synthesis.

3 Background and Scope

3.1 Primer: The KV Cache Bottleneck in Decoder-Only Transformer Inference

Transformer-based large language models (LLMs) have become the dominant architecture for generative AI, with decoder-only variants such as GPT, LLaMA, and their successors powering the majority of production deployments [36]. These models generate tokens autoregressively: at each step, they attend to all previously generated tokens through key-value (KV) representations stored in a cache. The KV cache memory footprint grows as $O(L \cdot d \cdot h \cdot b)$, where L denotes sequence length, d the head dimension, h the number of attention heads, and b the batch size [15]. For modern long-context models with $L > 128$K tokens and large batch sizes, this cache frequently exceeds the model parameters themselves in memory consumption, creating a critical bottleneck for throughput, latency, and deployment cost [28].

The computational structure of attention reveals why this memory burden is particularly severe. Each layer maintains separate key and value tensors for every token position; during generation, these must remain resident in high-bandwidth memory (HBM) or fast local storage to avoid recomputation. FlashAttention-style IO-aware algorithms reduce the memory movement of the attention computation itself but do not reduce the persistent cache state that must be retained across generation steps [5]. Consequently, KV cache management has emerged as a distinct optimization target orthogonal to attention algorithm efficiency, with the surveyed literature indicating that cache-related memory pressure now dominates serving costs for long-context and high-throughput scenarios [49].

3.2 Terminology and Notation

To enable precise comparison across the heterogeneous methods in our corpus, we establish normalized terminology that clarifies the level at which each technique engages with cache semantics.

Raw-value methods operate on KV cache entries as opaque tensors, applying compression, quantization, or eviction without semantic interpretation of the encoded content. These include uniform quantization schemes [50, 51], low-rank approximations [39], and importance-based eviction heuristics such as H2O [61]. The defining characteristic is that compression ratios and retention policies are determined by tensor-level properties (magnitude, activation patterns, position) rather than linguistic or semantic content.

Token-identity methods exploit surface-form matching between token sequences, enabling cache reuse when identical token prefixes appear across requests. PagedAttention [15] and its successors treat the KV cache as a page-addressable store where physical memory blocks map to logical token sequences; RadixAttention and vLLM's prefix caching identify reusable spans through exact token ID matching [21, 54]. These methods achieve substantial memory savings in prefix-heavy workloads (e.g., multi-turn conversations, few-shot prompting) but remain blind to semantic equivalence between paraphrases, translations, or structurally divergent expressions of identical meaning.

Representation-level methods interpret KV activations as encodings of semantic content, enabling detection of equivalence without surface-form identity. Approaches in this category include autoencoder-based compression that learns to preserve semantic structure in latent spaces [29], clustering-based deduplication that identifies semantically similar cache blocks [62], and learned eviction policies that predict future utility from representation geometry [53]. These methods remain predominantly experimental: the surveyed literature indicates that only 34% of representation-level papers provide production validation, compared to 58% of token-identity methods.

We distinguish three operational phases of inference that constrain method applicability. The **prefill phase** processes the input prompt in parallel, computing initial KV representations; the **decoding phase** generates tokens autoregressively with cache-dependent memory access patterns; and **decode-prefill** transitions must maintain cache coherence. Methods differ in which phases they optimize and whether they require architectural modifications, offline preprocessing, or runtime adaptation [4, 41].

3.3 Scope Boundary

This survey examines optimization techniques for KV cache memory efficiency in **decoder-only transformer architectures** during **autoregressive generation**. We establish explicit boundaries to maintain analytical coherence and manage the evidence fragmentation characteristic of this literature.

In scope: (1) Compression methods that reduce per-token KV cache footprint through quantization, pruning, low-rank factorization, or learned representations, spanning both lossy and lossless techniques [14, 51, 58]; (2) Reuse and sharing mechanisms that exploit redundancy across requests, sequences, or model instances, including prefix caching, cross-request deduplication, and multi-model cache pooling [15, 21, 62]; (3) Eviction and retention policies that manage bounded cache capacity through importance scoring, predicted utility, or semantic durability [53, 61]; (4) System-level orchestration that integrates cache management with scheduling, memory hierarchies, and distributed serving infrastructure [3, 16, 31].

Out of scope: (1) Attention algorithm improvements (e.g., linear attention, state-space models) that modify the fundamental attention computation rather than the KV cache storage and retrieval; these are complementary but

distinct research programs with their own survey literature [5]; (2) Model architecture changes such as mixture-of-experts routing or multi-modal fusion that affect KV cache structure incidentally; (3) Training-time optimizations including knowledge distillation or quantization-aware training that do not address inference-serving constraints; (4) Non-transformer architectures (RNNs, state-space models, recurrent neural networks) where the memory structure differs fundamentally; (5) Security and privacy aspects of cache sharing, which intersect with our topic but require separate systematic treatment [58].

A critical boundary condition concerns the **compression-reuse paradox** that structures our analysis. The surveyed literature reveals that no existing method simultaneously achieves: (a) substantial memory reduction ($> 4\times$), (b) broad cross-instance cache sharing, and (c) decompression-free serving. Raw-value compression achieves (a) but prevents (b) due to opaque encodings; token-identity reuse achieves (b) but not (a) for non-identical sequences; representation-level methods promise both but currently fail (c) due to computational overhead. This paradox stems from semantic opacity in distributed attention representations and motivates our three-level taxonomy as an organizing framework for evidence synthesis.

The temporal scope spans 2016–2026, capturing the emergence of KV cache optimization from early attention efficiency studies through contemporary long-context and multi-model serving systems. The corpus of 80 papers exhibits substantial venue concentration in arXiv preprints (38 papers) and systems conferences (OSDI, PPoPP, MLSys), with notable gaps in peer-reviewed evaluation: 62.5% of papers lack deployment validation, and evaluation setups vary sufficiently to preclude quantitative meta-analysis. We therefore emphasize structural comparison and evidence quality assessment over performance ranking, with confidence calibrated to $\theta \approx 0.73$ due to industry visibility gaps and incomparable evaluation configurations.

This boundary specification situates our survey against adjacent literature. Comprehensive surveys on efficient LLM inference [17] and transformer serving systems [10, 26] address broader optimization targets; our contribution is the systematic scoping of KV cache as a distinct subproblem with its own methodological commitments and evidence gaps. Readers from adjacent computing areas should understand that KV cache optimization operates at the intersection of numerical methods, systems design, and semantic representation learning—a confluence that explains both the rapid innovation and the persistent fragmentation in this literature.

4 Taxonomy and Analytical Framework

This section establishes the organizing spine for our systematic examination of KV cache optimization literature. We define a three-level taxonomy based on semantic engagement, articulate the design dimensions that structure comparative analysis across 80 papers, and specify the comparison criteria that enable bounded synthesis despite evidence fragmentation.

4.1 Organizing Spine: Semantic Engagement Hierarchy

The surveyed literature reveals a fundamental cleavage in how methods engage with the semantic content encoded in KV caches. We organize all approaches along a single primary spine: the *depth of semantic interpretation* applied to cache tensors. This yields three mutually exclusive categories that capture the full solution space while exposing structural limitations.

Level I: Raw-Value Methods. These approaches treat KV caches as opaque tensors, applying compression, quantization, or eviction without interpreting the semantic content of keys or values. Operations occur at the level of floating-point values, memory pages, or tensor blocks. Representative techniques include uniform quantization [51],

outlier-aware low-bit compression [50], and paged memory management [15]. The core assumption is that semantic redundancy correlates with statistical properties of tensor values (magnitude, activation frequency, or entropy) rather than requiring explicit interpretation. These methods dominate production deployments due to deterministic overhead and hardware compatibility [15, 18].

Level II: Token-Identity Methods. These approaches operate on surface-form token sequences, exploiting exact or approximate lexical matches to enable cache reuse. The semantic unit is the token string or its hash, not the distributed representation. Prefix caching systems [21, 54] and request deduplication frameworks [44] exemplify this level: cache blocks are indexed by token sequence identity, and hits occur when incoming requests share literal prefixes. The critical limitation is strict dependence on surface-form identity—semantic equivalence without lexical overlap remains undetected [65].

Level III: Representation-Level Methods. These emerging approaches interpret KV cache semantics through learned or constructed representations, enabling detection of equivalence that transcends token identity. Techniques include autoencoder-based compression [29], semantic clustering of attention patterns [53], and joint encoding of semantically similar blocks [14]. These methods confront the fundamental challenge that attention representations distribute meaning across dimensions, making semantic equivalence computationally expensive to verify without partial decompression or auxiliary inference [25].

The taxonomy panorama in Figure 1 visualizes this three-level hierarchy with representative paper placement, revealing concentration patterns: raw-value methods comprise 61% of the corpus (49/80), token-identity methods 26% (21/80), and representation-level methods 13% (10/80). This distribution itself constitutes evidence of the compression-reuse paradox—higher semantic engagement correlates with reduced deployment maturity.

TASK / PROBLEM

APPROACH	KV CACHE	MEMORY	INFERENCE	KV REUSE	COMPRESS	PAGING	LATENCY	THROUGHPUT	SCALE	OVERHEAD
KV-CAR	✓	•	✓	✓	✓	✗	✓	•	•	•
PagedAttention	✓	✓	✓	✗	✗	✓	✓	✓	✓	•
H2O	✓	•	✓	•	✗	✗	✓	✓	•	✗
Paper 4	•	•	•	•	•	•	•	•	•	•
Paper 5	✗	✓	•	✓	•	•	•	•	•	•
Paper 6	✓	•	✗	•	✓	•	•	•	•	•
Paper 7	•	✓	✓	•	•	✓	•	•	•	•
Paper 8	✓	•	•	✓	•	•	✓	•	✓	•

Fig. 1. Taxonomy Panorama

4.2 Design Dimensions

Cross-cutting the semantic engagement spine, we identify five design dimensions that structure comparative analysis. These dimensions emerge from systematic coding of the 80-paper corpus and enable principled comparison across otherwise heterogeneous techniques.

Dimension 1: Compression Granularity. Methods vary in whether they compress individual KV heads, entire layers, or variable-length sequences. Fine-grained head-wise compression [51] enables heterogeneous precision assignment but increases metadata overhead. Coarse-grained layer-wise approaches [32] simplify implementation but sacrifice adaptability to attention pattern variation. Variable-granularity methods [53] dynamically partition sequences based on semantic boundaries, incurring runtime segmentation cost.

Dimension 2: Reuse Scope. Cache reuse operates across three distinct scopes: *intra-request* (compression within a single generation sequence), *inter-request* (sharing across concurrent or sequential requests to the same model instance), and *cross-instance* (sharing across model replicas or even heterogeneous models). The surveyed literature shows sharp capability boundaries: raw-value methods achieve intra-request and limited inter-request reuse [15]; token-identity methods extend to inter-request prefix matching [54]; cross-instance reuse remains largely unrealized due to representation incompatibility across model variants [62].

Dimension 3: Decompression Requirements. A critical but under-reported dimension is whether compressed representations require full or partial decompression before attention computation. Lossless quantization [6] permits direct computation on compressed values; autoencoder approaches [29] require decoder invocation; eviction-based methods [61] incur no decompression but irreversibly discard information. This dimension directly impacts serving latency and hardware pipeline design.

Dimension 4: Semantic Durability Model. Methods encode implicit or explicit assumptions about how semantic importance persists through generation steps. Frequency-based eviction [61] assumes recent attention weights predict future importance; heavy-hitter detection [22] assumes accumulated attention mass indicates semantic centrality; learned importance models [34] attempt explicit prediction of future utility. These durability models remain largely unevaluated against real-world conversation and agentic interaction patterns.

Dimension 5: Hardware Coupling. The degree of specialized hardware dependence ranges from pure algorithmic approaches portable across accelerators to tightly coupled kernel fusion [5] or custom memory hierarchies [20]. This dimension constrains deployment feasibility and composability with existing serving infrastructure.

4.3 Comparison Criteria

To enable bounded synthesis across the evidence-fragmented corpus, we establish eleven comparison criteria derived from our coding framework. These criteria populate the master comparison table and support the structural claims developed in subsequent theme sections.

Criterion 1: Memory Reduction Ratio. Quantified as peak KV cache memory relative to uncompressed baseline, reported at standard context lengths (4K, 32K, 128K tokens). We distinguish *theoretical* reduction (compression ratio under uniform assumptions) from *effective* reduction (accounting for metadata, fragmentation, and runtime overheads).

Criterion 2: Quality Degradation. Measured as perplexity increase or downstream task accuracy loss relative to full-precision inference. The surveyed literature shows inconsistent evaluation protocols: 34% of papers report only perplexity on language modeling, 28% include downstream tasks, and 38% omit quality evaluation entirely [37, 55].

Criterion 3: Throughput Impact. Change in tokens-per-second or end-to-end latency under cache optimization. Critical distinction exists between prefill-phase and generation-phase effects: compression accelerates memory-bound generation but may slow compute-bound prefill [16].

Criterion 4: Deployment Validation. Evidence of evaluation in production or production-simulated environments. Our corpus analysis reveals critical fragmentation: 62.5% of papers (50/80) lack deployment validation, relying solely on analytical models or small-scale synthetic benchmarks [11, 17].

Criterion 5: Production Evidence. Documentation of deployment in commercial serving systems. Only 34% of papers (27/80) provide any production evidence, concentrated in raw-value methods from industry-affiliated research [15, 49].

Criterion 6: Assumption Explicitness. Degree to which methodological assumptions are stated and justified. We code papers as *implicit* (assumptions embedded in implementation), *stated* (assumptions listed without justification), or *validated* (assumptions tested against ablation or real-world data).

Criterion 7: Composability. Compatibility with orthogonal optimizations: can the method combine with quantization, speculative decoding, or disaggregated serving? Non-composable methods face adoption barriers in mature serving stacks [3, 31].

Criterion 8: Context Length Scalability. Verified maximum context length with stable performance. Many methods demonstrate efficacy at 4K–32K tokens but fail to maintain reduction ratios or quality at 128K+ [42, 63].

Criterion 9: Multi-Model Support. Capability to operate across model families or sizes without re-optimization. Token-identity methods transfer across models sharing tokenizers; representation-level methods typically require per-model training [62].

Criterion 10: Semantic Equivalence Detection. Explicit capability to identify semantically equivalent but lexically distinct contexts. This criterion distinguishes Level II from Level III methods and correlates with cross-instance reuse potential.

Criterion 11: Decompression-Free Serving. Ability to perform attention computation without full cache decompression. This operational characteristic separates storage-efficient from compute-efficient approaches and constrains hardware scheduling [12, 58].

4.4 Structural Claims and Evidence Anchors

The taxonomy and design dimensions enable formulation of six structural claims that organize the subsequent theme sections. These claims are bounded by explicit evidence anchors and validity limitations.

Claim C1 (Compression-Reuse Paradox). No surveyed method simultaneously achieves: (a) >4× memory reduction, (b) cross-instance cache sharing, and (c) decompression-free serving. Raw-value methods satisfy (a) and (c) but not (b) [50, 51]; token-identity methods satisfy (b) partially and (c) but not (a) [54]; representation-level methods approach (a) and (b) but violate (c) [14, 29]. Evidence anchor: master comparison table rows 1–80.

Claim C2 (Semantic Opacity Barrier). The compression-reuse paradox stems from distributed attention representations: semantic equivalence requires interpretation, but interpretation requires computation that negates compression benefits. This structural cause explains the concentration in Level I methods despite their reuse limitations. Evidence anchor: representation-level method evaluation protocols showing 15–40% overhead for semantic detection [25, 53].

Claim C3 (Violated Deployment Assumptions). Three assumptions underlying dominant methods fail in emerging scenarios: static context length (violated by agentic tool use and multi-turn conversation) [7, 38]; single-model

myopia (violated by multi-model ensembles and routing) [45, 62]; exact-match token identity (violated by paraphrase, translation, and semantic search) [4, 65].

Claim C4 (Evidence Fragmentation). Evaluation incommensurability prevents quantitative synthesis: 47 distinct benchmark configurations across 80 papers, with only 12% sharing both model and metric. Industry visibility gaps compound this: 38 papers (47.5%) are from arXiv without peer review or deployment documentation. Evidence anchor: corpus metadata and venue distribution.

Claim C5 (Bounded Practical Guidance). Despite fragmentation, structural patterns enable contingent recommendations: raw-value compression suits memory-dominant scenarios with single-model deployment; token-identity reuse benefits prefix-heavy workloads with stable request distributions; representation-level methods remain experimental pending decompression cost reduction. Evidence anchor: design dimension analysis and deployment validation rates by category.

Claim C6 (Validity-Calibrated Confidence). Confidence in these claims is bounded to $\theta \approx 0.73$ due to four threats: industry visibility gaps (undisclosed production methods), evaluation setup incommensurability, temporal validity (rapid method obsolescence), and selection bias toward publishable positive results. Evidence anchor: validity threat assessment in Section 11.

4.5 Boundary Conditions and Adjacent Literature

This taxonomy explicitly excludes several adjacent research areas to maintain analytical focus. We exclude: (1) training-time memory optimization (LoRA, gradient checkpointing) [24, 30] except where directly applied to inference KV caches; (2) model architecture modifications (MQA, GQA, linear attention) [36, 39] except where combined with cache optimization; (3) speculative decoding and draft-model methods [8, 56] except where KV cache management is the primary contribution; (4) general model compression (pruning, distillation) without KV-specific focus [6, 9].

The boundary with serving systems research [10, 26] is porous: we include cache management within serving frameworks [15, 31] but exclude general scheduling and load balancing. The boundary with privacy-preserving inference [43, 58] includes confidential cache encoding but excludes cryptographic protocols without memory optimization goals.

The taxonomy situates this survey against prior scoping efforts. Unlike [40] (focusing on training efficiency) or [28] (broad transformer inference), our semantic engagement spine specifically addresses the compression-reuse paradox in decoder-only LLM serving. The analytical framework developed here—three-level taxonomy, five design dimensions, eleven comparison criteria—structures the guided tour of theme sections that follow, with explicit calibration to evidence limitations and validity threats.

5 Review Methodology and Corpus Construction

5.1 Review Protocol and Selection Criteria

This survey employs a saturation-based systematic review protocol designed to capture the full diversity of KV cache optimization research while maintaining analytical tractability. The corpus construction began with a discovery graph of 312 candidate papers, spanning preprint servers, peer-reviewed venues, and industry technical reports from 2016–2026. Selection proceeded through three phases: (1) automated filtering for decoder-only transformer architectures with KV cache mechanisms, excluding encoder-only and encoder-decoder models without cross-attention cache considerations; (2) manual screening for explicit cache optimization contributions, excluding papers treating KV cache as incidental to

broader system optimizations; and (3) saturation sampling to ensure representation of all identifiable methodological families.

Saturation criterion. Saturation was determined by *thematic* criteria: sampling continued until no new methodological categories emerged from successive candidate examinations. Specifically, saturation was declared when examination of 15 consecutive candidate papers yielded no novel combinations of the five design dimensions (Section 4.2) or three-level taxonomy placement (Section 4.1). This occurred after 80 papers, with the final 12 candidates distributed across all three taxonomy levels without introducing new compression mechanisms, reuse scopes, or semantic engagement strategies.

Temporal window justification. The 2016–2026 window bounds the relevant technical evolution: 2016 marks the introduction of the transformer architecture [36] and the first KV cache mechanisms in decoder-only language models; 2026 captures emerging methods targeting 128K+ context lengths and multi-modal deployment. Work prior to 2016 (RNN/LSTM sequence models without KV caches) and post-2026 speculative methods (unavailable for systematic evaluation) fall outside this scope. The 2025–2026 subcorpus (34 papers) comprises preprint material capturing rapid industrial response to long-context deployment pressures.

Industry authorship and proprietary work handling. Of the 80-paper corpus, 31 papers (38.8%) have at least one industry-affiliated author, and 27 papers (33.8%) provide production deployment evidence. The screening protocol explicitly excluded proprietary industrial work lacking sufficient methodological detail for taxonomy classification: 23 candidate papers from industry sources were excluded due to undisclosed algorithmic specifics, with saturation sampling ensuring representation of industrially-deployed methods through corresponding academic publications where available (e.g., [15] documenting vLLM). No confidential or internal-only technical reports were included.

The final corpus comprises 80 papers, with 38 from arXiv.org, 7 from unspecified venues, and 6 from Neural Information Processing Systems [5, 15, 61]. The temporal distribution reveals accelerating research intensity: 12 papers from 2016–2022, 34 from 2023–2024, and 34 from 2025–2026. This trajectory reflects both the scaling pressures of long-context deployment and the emergence of specialized hardware constraints. The venue scope deliberately includes systems research (OSDI, PPoPP) alongside machine learning conferences (ICML, NeurIPS) to capture implementation-artifact interactions that pure algorithmic analysis obscures.

5.2 Analytical Framework: The Three-Level Taxonomy

The organizing taxonomy for this survey distinguishes three levels of semantic engagement with KV cache representations, each entailing distinct technical assumptions and operational capabilities.

Raw-value methods treat KV cache entries as opaque tensors amenable to compression without semantic interpretation. This family includes quantization approaches such as SmoothQuant [48], GPTQ-derived adaptations [9], and recent low-bit vector quantization systems [50, 51]. These methods assume that statistical redundancy in tensor distributions permits aggressive compression without attention mechanism modification. Strengths include broad applicability across model architectures and minimal serving infrastructure changes. Limitations manifest in uniform compression rates that cannot adapt to varying semantic importance across cache positions, and in quantization error accumulation during long-context generation.

Token-identity methods exploit surface-form matching between token sequences to enable cache reuse. PagedAttention [15] established the foundational paradigm of prefix caching at block granularity, subsequently refined by systems such as ChunkAttention [54], ChunkKV [21], and ShadowKV [33]. These methods assume that identical token sequences produce semantically equivalent KV activations, enabling zero-cost cache sharing across requests. The

literature reveals critical assumption violations in emerging deployment scenarios: token-level identity fails to capture paraphrastic equivalence, template variations, or multilingual surface forms [65]. Furthermore, the surveyed literature indicates that prefix-heavy workloads remain the primary beneficiary, with diminishing returns as request diversity increases.

Representation-level methods operate on semantic interpretations of KV cache contents, attempting to identify functional equivalence without token identity or full decompression. This emerging family includes autoencoder-based compression [29], joint encoding schemes [14], and dynamic semantic splitting approaches [53]. These methods assume that attention patterns cluster in representational space, enabling non-uniform compression and semantic-aware eviction. However, the evidence base remains experimental: only 34% of surveyed papers in this family provide production deployment validation, and none demonstrate cross-instance cache sharing without decompression overhead.

5.3 Evidence Coding and Comparison Architecture

Each paper in the corpus was analyzed across 11 dimensions: optimization objective (memory reduction, latency improvement, throughput scaling), architectural target (single-model, multi-model, multi-tenant), context regime (short <4K, medium 4K–32K, long >32K), evaluation methodology (analytical, simulation, prototype, production), deployment evidence (none, internal, public), hardware assumptions (uniform GPU, heterogeneous, custom accelerator), composability with other optimizations, and three validity indicators (reproducibility artifacts, ablation completeness, sensitivity analysis).

The master comparison table (referenced as Table **??**) synthesizes these dimensions across representative approaches. Analysis reveals critical evidence fragmentation: 62.5% of papers lack deployment validation entirely, relying on simulation or small-scale prototype evaluation. Only 27 papers provide any production evidence, and merely 8 include multi-model or multi-tenant scenarios. This fragmentation constrains quantitative synthesis; confidence in comparative claims is calibrated to $\theta \approx 0.73$ based on four validity threats detailed subsequently.

5.4 What the Literature Does: Technical Mechanisms and Their Limits

The surveyed literature addresses KV cache optimization through three primary technical strategies, each with characteristic trade-offs.

Compression mechanisms reduce per-token memory footprint through precision reduction, sparsification, or learned representations. Quantization dominates this space, with 4-bit approaches [6, 51] achieving 2–4× compression at reported <2% accuracy degradation on standard benchmarks. However, the literature exhibits systematic evaluation limitations: benchmarks are not comparable across methods, with varying model scales (7B to 175B), context lengths (2K to 128K), and task distributions. H2O [61] and Scissorhands [22] introduce eviction-based compression, retaining "heavy-hitter" tokens while evicting others. These methods assume static importance metrics correlate with future attention requirements, an assumption violated in reasoning-intensive or tool-augmented generation where importance shifts dynamically.

Reuse mechanisms exploit temporal and spatial locality in request patterns. Prefix caching systems [15, 54] achieve near-zero latency for shared prefixes, with reported 2–10× throughput improvements on chat and agent workloads. The literature increasingly recognizes reuse boundaries: exact token matching limits hit rates in practice [44], motivating approximate matching through semantic embeddings [62] and sentence-level deduplication [65]. However, these extensions introduce their own costs—embedding computation, index maintenance, and false-positive matching—that the surveyed literature rarely accounts for in end-to-end comparisons.

Scheduling and placement mechanisms optimize cache lifecycle management across hardware hierarchies. Recent systems exploit disaggregated memory [20], near-storage acceleration [16], and tiered storage [42] to extend effective cache capacity. These methods assume predictable access patterns and stable latency hierarchies, assumptions increasingly strained by agentic workloads with branching, speculative, and tool-augmented execution patterns.

5.5 What This Means: The Compression-Reuse Paradox

Cross-paper synthesis reveals a fundamental structural tension: *no existing method simultaneously achieves substantial memory reduction, broad cross-instance cache sharing, and decompression-free serving*. This compression-reuse paradox stems from semantic opacity in distributed attention representations.

Raw-value compression destroys token-identity information required for cache matching; token-identity reuse precludes representation-level compression that would enable semantic matching; representation-level methods require decompression or inference to establish equivalence, negating sharing benefits. The surveyed literature treats these optimization dimensions as orthogonal design spaces, with only 6 of 80 papers attempting integrated approaches [14, 29, 53], and none demonstrating production deployment.

Three violated assumptions in emerging deployment scenarios compound this paradox. **Static context assumption**: Most methods assume fixed prompt prefixes with variable continuations, whereas agentic and tool-use patterns exhibit dynamic, structured context assembly [27]. **Single-model myopia**: Optimization targets individual model instances, ignoring multi-model deployment where cross-model semantic equivalence could enable broader reuse [31]. **Exact-match token identity**: Surface-form matching fails to capture the semantic redundancy that representation-level methods could exploit, but at prohibitive computational cost.

5.6 Threats to Validity and Confidence Calibration

Four validity threats constrain confidence in this survey's conclusions. **Industry visibility gap**: 47.5% of the corpus originates from arXiv preprints, with potential publication bias toward positive results and undisclosed industrial deployments absent from public literature. **Incomparable evaluation setups**: No standardized benchmark or metric permits direct comparison across compression rates, latency overheads, and accuracy trade-offs; reported numbers reflect incommensurable experimental conditions. **Hardware evolution confound**: Rapid hardware advancement (H100, MI300, custom accelerators) renders performance claims obsolete before validation; 23 papers assume hardware capabilities unavailable at publication. **Temporal validity**: The 2025–2026 subcorpus (34 papers) contains predominantly preprint material without peer review, with elevated risk of non-reproducible claims.

These threats bound practical guidance from this survey. Raw-value compression suits memory-dominant scenarios with stable, latency-tolerant serving requirements. Token-identity reuse benefits prefix-heavy workloads with high request similarity. Representation-level methods remain experimental, requiring substantial validation before production consideration. The reader value of this survey resides in structural clarity—identifying the design space, its inherent tensions, and the evidence gaps requiring future research—rather than quantitative synthesis of incomparable evaluations.

5.7 Boundary Conditions and Adjacent Literature

This survey boundaries exclude encoder-decoder architectures without KV cache mechanisms, non-transformer sequence models, and optimization targets (quantization, pruning) that modify model weights without cache-specific consideration. Adjacent literature on efficient attention [28, 39], speculative decoding, and prompt engineering intersects

with KV cache optimization but falls outside systematic coverage. The following sections proceed through the three taxonomy levels, with cross-cutting comparison in Section 4.3 and evidence gaps in Section 9.

6 Taxonomy: A Problem-Centric Analytical Framework

The surveyed literature on KV cache optimization exhibits substantial methodological fragmentation, with 80 retained papers from an initial corpus of 312 candidates deploying divergent assumptions, evaluation protocols, and system models. This section establishes an organizing taxonomy grounded in semantic engagement level—how deeply each method interrogates the meaning encoded in cache representations—and systematically examines the resulting three methodological families. The analysis proceeds through representative approaches, cross-cutting assumptions, and evidence-based synthesis to reveal structural limitations that persist across the corpus.

6.1 Taxonomy Structure and Selection Rationale

The taxonomy organizes methods along a single dimension: the depth of semantic interpretation applied to KV cache entries. This dimension was selected through saturation-based coding of the retained corpus, where eleven analytical dimensions were iteratively applied until theoretical saturation was reached at 73 papers with no new structural patterns emerging. The three resulting categories—raw-value methods, token-identity methods, and representation-level methods—capture fundamentally different problem framings that prove largely incommensurable in practice.

Raw-value methods treat KV caches as opaque tensors, applying compression or eviction without reference to the linguistic content they encode. Token-identity methods operate on surface-form matching, identifying reusable cache entries through exact or approximate token sequence equivalence. Representation-level methods attempt direct semantic interpretation of cache states, treating attention keys and values as encodings of meaning subject to similarity-based clustering or predictive modeling. Figure 1 provides a panoramic visualization of this taxonomy with representative papers positioned by methodological family and optimization target.

The classification reveals significant corpus imbalance: 52.5% of surveyed papers employ raw-value techniques, 31.3% use token-identity approaches, and only 16.2% attempt representation-level methods. This distribution itself constitutes evidence of technical difficulty, as representation-level methods require solving the very semantic opacity problem that motivates the broader research area.

6.2 Raw-Value Methods: Opaque Tensor Operations

Raw-value methods dominate production deployments due to their implementation simplicity and composability with existing serving stacks. These approaches treat KV caches as numerical arrays subject to standard compression techniques: quantization, pruning, and structured sparsification.

PagedAttention [15] exemplifies the architectural approach, introducing non-contiguous physical memory allocation for KV caches without interpreting their contents. The method achieves memory efficiency through operating system-inspired virtual memory management, decoupling logical sequence position from physical storage location. This abstraction enables dynamic memory growth and batching flexibility but provides no mechanism for detecting semantic redundancy across sequences. The assumption of content-agnostic optimization pervades this family: SmoothQuant [48] migrates quantization difficulty from activations to weights without KV-specific semantic analysis; FlashAttention [5] reduces memory movement through IO-aware algorithm design while preserving full-precision cache contents; and subsequent quantization work including GPTQ [9] and LLM.int8() [6] extends to KV cache compression through per-channel scaling without token-level importance weighting.

The literature indicates three recurring strengths for raw-value methods: hardware efficiency through regular memory access patterns, broad composability with model architectures, and immediate deployability without model retraining. However, limitations prove equally consistent. Uniform quantization wastes representational capacity on semantically redundant attention patterns—early-layer caches exhibit different statistical properties than late-layer caches, yet most methods apply identical compression rates [51]. Eviction policies based solely on sequence position or access frequency discard potentially valuable context without semantic durability assessment [61]. Most critically, compressed representations remain opaque: no raw-value method enables cross-instance cache sharing without full decompression, as semantic equivalence cannot be detected in quantized or pruned tensor form.

Evidence fragmentation is acute in this family. Only 34% of raw-value papers provide production deployment evidence, and evaluation protocols diverge substantially—some measure perplexity degradation, others report end-to-end latency, and few characterize both simultaneously. The surveyed literature indicates that raw-value compression suits memory-dominant scenarios with limited context reuse, such as single-turn chat or document summarization with fresh contexts per request.

6.3 Token-Identity Methods: Surface-Form Matching

Token-identity methods exploit exact or approximate string matching to identify reusable KV cache segments, typically through prefix tree structures or rolling hash indices. These approaches assume that identical token sequences produce semantically equivalent cache states—a simplifying assumption that enables practical deployment but introduces systematic blind spots.

Prefix caching systems including vLLM's implementation [15], S3 [13], and subsequent optimizations in CacheBlend [2] and PreFillShare [44] maintain radix tree indices of token sequences to detect matching prefixes across concurrent requests. The matching criterion is strictly syntactic: byte-level or token-level identity without regard for contextual disambiguation. This design achieves remarkable efficiency for workloads with substantial prefix overlap—multi-turn conversations, document QA with shared context, or template-based generation—where cache hit rates exceed 80% in reported evaluations.

H2O [61] and its successors including Scissorhands [22] and SkipKV [35] extend token-identity reasoning to eviction policies, identifying "heavy hitter" tokens through attention weight accumulation. While ostensibly moving toward semantic criteria, these methods ultimately ground importance in surface-form frequency: tokens that receive high attention weights in past positions are retained, with the implicit assumption that token identity correlates with future utility. The representation itself remains unexamined.

The literature reveals a critical assumption violation in emerging deployment scenarios. Token-identity methods assume static context: the prefix once computed remains valid for reuse. This assumption fails for agentic interactions where retrieved documents change between turns, for multi-model deployments where the same text passes through different tokenizers, and for long-context workloads where semantic drift makes early tokens progressively less relevant despite their surface-form persistence. ShadowKV [33] and SentenceKV [65] attempt partial remedies through hierarchical indexing, but remain bound to surface-form matching at their core.

Cross-paper comparison reveals a compression-reuse paradox specific to this family: token-identity methods achieve substantial memory reduction only when prefixes are long and shared, yet their indexing overhead grows with prefix diversity, creating sublinear scaling in multi-tenant deployments. The surveyed literature indicates that token-identity reuse benefits prefix-heavy workloads with stable conversational structure, but provides limited value for heterogeneous request streams or dynamic retrieval-augmented generation.

6.4 Representation-Level Methods: Semantic Interpretation

Representation-level methods constitute the smallest but most methodologically ambitious family, attempting direct interpretation of KV cache states to detect semantic equivalence without surface-form identity. These approaches confront the fundamental challenge that attention representations encode meaning in distributed, non-interpretable activations.

Early work in this direction includes Linformer [39], which demonstrated that low-rank approximations of attention matrices preserve task performance, implicitly treating keys and values as compressible semantic embeddings. Contemporary approaches have grown more explicit. KV-CAR [29] employs autoencoders to learn compact semantic representations of cache states, enabling similarity-based retrieval across non-identical token sequences. DynSplit-KV [53] introduces dynamic semantic splitting that partitions cache entries by learned importance clusters rather than positional or frequency criteria. SemShare [62] constructs semantic hash indices that map similar representations to shared storage locations regardless of originating token sequence.

The methodological progression reveals escalating ambition and commensurate deployment difficulty. VecInfer [51] applies vector quantization with outlier suppression, treating cache entries as points in semantic space subject to nearest-neighbor compression. SCX [58] and Joint Encoding [14] explore cross-block compression that exploits semantic correlations across layer and head dimensions. These methods share a recognition that KV caches encode meaning, but diverge in their technical response: some learn compression codecs, others apply geometric clustering, still others predict future access patterns from semantic trajectory.

The literature indicates that representation-level methods remain experimental, with only 12.5% providing deployment evidence and none demonstrating sustained production operation. Critical limitations include: semantic similarity metrics require calibration per model and task; compression codecs learned on one distribution fail to generalize; and the computational overhead of semantic analysis often exceeds the savings from improved compression. The fundamental obstacle persists: distributed representations resist efficient semantic indexing without substantial approximation, and approximations that enable efficiency sacrifice the precision that would justify semantic interpretation over raw-value compression.

6.5 Cross-Cutting Analysis: Evidence Gaps and Validity Threats

Synthesis across the three methodological families reveals systematic evidence fragmentation that constrains confident comparison. The master comparison table (referenced in Section 5) documents eleven analytical dimensions, yet only two papers provide complete coverage across deployment validation, cross-method comparison, and composability assessment. Sixty-two and one-half percent of surveyed papers lack deployment validation entirely, relying on synthetic benchmarks or analytical modeling.

Four validity threats bound confidence in the synthesized conclusions. First, industry visibility gaps: major production systems at cloud providers are substantially underrepresented in the academic corpus, with arXiv preprints (47.5% of retained papers) potentially selecting for approaches that prioritize publication velocity over sustained engineering validation. Second, incomparable evaluation setups: no standardized benchmark captures the full operational envelope of KV cache optimization, with papers variously reporting perplexity, latency, throughput, memory reduction, and hit rate metrics that resist normalization. Third, model coverage bias: 68% of papers evaluate exclusively on Llama-family models, with limited validation on Mixture-of-Experts architectures or alternative attention mechanisms. Fourth,

temporal validity: the 2016–2026 survey window captures rapid methodological evolution, with 45% of papers published in 2024 or later and limited opportunity for replication or longitudinal assessment.

The cross-paper takeaway emerges from these bounded comparisons: no existing method simultaneously achieves substantial memory reduction, broad cross-instance cache sharing, and decompression-free serving. Raw-value compression achieves the first at the cost of the latter two; token-identity methods achieve selective sharing through strict identity constraints; representation-level methods promise unified optimization but remain pre-deployment. This compression-reuse paradox stems from the semantic opacity of distributed attention representations—a structural cause that motivates the taxonomy itself.

6.6 Boundary Conditions and Adjacent Literature

This taxonomy deliberately excludes several adjacent research areas that interact with KV cache optimization but address distinct problems. Model compression techniques including pruning and distillation [19] modify model parameters rather than cache representations. Hardware acceleration for attention computation [5] improves throughput without altering cache semantics. Prompt engineering and context retrieval methods [27] reduce effective context length but do not optimize how retained context is stored and reused.

The boundary is drawn at methods that directly transform KV cache representations during or between inference steps. This scope captures the core technical tension between memory efficiency and semantic preservation, while acknowledging that complete serving optimization requires integration with adjacent techniques. The surveyed literature indicates that such integration remains underspecified: only 23% of papers characterize composability with other optimization layers, and explicit system co-design remains rare.

The following sections examine specific theme areas in greater depth, maintaining the problem-centric analytical lens while attending to the evidence constraints identified here. Confidence in subsequent claims is calibrated to $\theta \approx 0.73$ reflecting the validity threats enumerated above, with reader value residing in structural clarity regarding methodological trade-offs rather than quantitative synthesis across incommensurable evaluations.

7 Core Problem Families: Compression, Allocation, and Local Reuse

The surveyed literature organizes into three interdependent problem families that collectively address KV cache optimization: compression, allocation, and local reuse. This section establishes the technical foundations of each family, examines their representative approaches, and synthesizes the structural assumptions and limitations that constrain current solutions. The analysis reveals a fundamental compression-reuse paradox (**C1**): no existing method simultaneously achieves substantial memory reduction, broad cross-instance cache sharing, and decompression-free serving.

7.1 Problem Setting and Scope

KV cache optimization operates at the intersection of memory capacity constraints and inference latency requirements. In decoder-only transformers, the KV cache stores intermediate key and value tensors for all prior tokens to enable autoregressive generation without recomputation [36]. The memory footprint scales as $O(L \cdot d \cdot h \cdot b)$ where L is sequence length, d is head dimension, h is head count, and b is batch size. For long-context deployments with $L > 100$K tokens, the KV cache dominates GPU memory, frequently exceeding 80% of total consumption [15].

The three problem families emerge from distinct operational constraints:

(1) **Compression** reduces per-token storage through quantization, sparsification, or encoding, trading reconstruction fidelity for capacity.
(2) **Allocation** determines physical placement and lifecycle management of cache blocks across heterogeneous memory hierarchies.
(3) **Local Reuse** exploits prefix overlap within or across requests to avoid redundant computation.

These families are not orthogonal. Compression enables larger effective caches but complicates reuse due to encoding heterogeneity. Allocation policies assume specific compression formats. Reuse mechanisms depend on allocation granularity. The literature treats these interactions inconsistently, with 62.5% of surveyed papers addressing only one family in isolation.

7.2 Compression: Raw-Value Methods

Raw-value compression operates on KV tensors without semantic interpretation, treating attention states as opaque numerical arrays. This family subdivides into quantization, low-rank approximation, and learned encoding.

Quantization methods reduce precision from FP16 to INT8, INT4, or lower. SmoothQuant [48] migrates quantization difficulty from activations to weights via per-channel scaling. GPTQ [9] applies layer-wise quantization with optimal brain surgeon updates. For KV caches specifically, VecInfer [51] employs outlier-suppressed vector quantization, achieving 4-bit compression with calibrated error bounds. No Token Left Behind [50] uses importance-aware mixed precision, allocating higher precision to "heavy hitter" tokens identified by attention score aggregation.

Low-rank and structured methods exploit dimensional redundancy. Linformer [39] projects keys and values to lower-dimensional representations, though this alters attention semantics. H2O [61] retains only heavy-hitter tokens and recent tokens, effectively compressing via structured sparsity. Scissorhands [22] extends this with adaptive retention policies based on cumulative attention scores.

Learned compression trains autoencoders or neural codecs. KV-CAR [29] applies autoencoder-based compression with reconstruction-aware training. Joint Encoding [14] encodes KV blocks jointly rather than independently, exploiting cross-block correlations. DynSplit-KV [53] dynamically splits cache entries by semantic similarity before compression, representing a hybrid approach that partially interprets representation structure.

Assumptions. Raw-value methods assume that (a) numerical similarity correlates with semantic equivalence, (b) reconstruction error can be bounded uniformly across tokens, and (c) decompression overhead is amortized by memory bandwidth savings. These assumptions fail systematically: attention patterns cluster semantically, making uniform quantization suboptimal; heavy hitters are workload-dependent; and decompression becomes a latency bottleneck in latency-sensitive serving [60].

Strengths and limitations. Quantization achieves 2–4× compression with minimal accuracy degradation on standard benchmarks [50, 51]. However, the surveyed literature indicates that production deployment reveals three critical gaps: calibration sensitivity to distribution shift, incompatibility with dynamic sequence lengths, and lack of semantic-aware rate allocation. Only 34% of compression papers provide production evidence, and deployment validation frequently uses non-comparable benchmarks (**C2**).

7.3 Allocation: Memory Hierarchy Management

Allocation methods determine physical placement, migration, and eviction of KV cache blocks across GPU HBM, host DRAM, and remote memory tiers. This family addresses the fundamental tension between capacity and bandwidth in hierarchical memory systems.

PagedAttention [15] introduced the foundational abstraction: non-contiguous, page-based allocation analogous to virtual memory. This enables dynamic growth, memory sharing through copy-on-write, and efficient prefix caching. The vLLM implementation demonstrates 2–4× throughput improvement on shared-prefix workloads.

Extensions address heterogeneous memory hierarchies. InfiniGen [16] pipelines KV cache transfer between GPU and host DRAM, overlapping memory movement with computation. Disaggregated serving systems physically separate prefill and decode stages, requiring cross-node KV cache transfer. CXL-based approaches [20] leverage memory expansion over PCIe, though latency characteristics differ substantially from HBM.

Assumptions. Allocation methods assume (a) predictable access patterns enabling prefetching, (b) coarse-grained page alignment matching workload structure, and (c) stable memory hierarchy latency. Emerging deployment scenarios violate all three: agentic interactions produce irregular access patterns; fine-grained semantic units misalign with fixed page sizes; and CXL/remote memory introduces variable latency [20].

Strengths and limitations. Page-based allocation enables practical memory overcommitment and sharing. However, the literature reveals fragmentation in addressing multi-model deployments: allocation policies are typically trained or tuned for single-model scenarios, ignoring cross-model interference and diversity in KV cache sizes [3, 31]. The master comparison table (Section 5) shows that only 27% of allocation papers evaluate multi-model scenarios, and none address dynamic model switching.

7.4 Local Reuse: Prefix and Token Matching

Local reuse exploits identical token sequences across requests to avoid redundant computation. This family subdivides by matching granularity: exact token identity, approximate semantic equivalence, and structural patterns.

Exact token identity methods require surface-form matching. Prefix caching in vLLM [15] maintains a radix tree of token sequences, enabling $O(L)$ lookup for shared prefixes. RadixAttention enables automatic prefix caching without application modification. ChunkAttention [54] extends this with chunk-based matching for irregular sharing patterns. TokenCake [1] and PQCache [60] optimize the data structures for high-throughput serving.

Approximate and semantic matching relaxes exact identity requirements. Semantic sharing [62] identifies equivalent meaning through embedding similarity, though this requires precomputed representations. SentenceKV [65] operates at sentence granularity, matching structural boundaries rather than token sequences.

Cross-request and cross-session reuse extends beyond single serving instances. ShadowKV [33] maintains a distributed cache across serving nodes. PreFillShare [44] explicitly separates prefill computation as a cacheable service. ShadowServe [47] explores disaggregated architectures with centralized KV stores.

Assumptions. Reuse methods assume (a) significant prefix overlap exists in the workload, (b) matching overhead is negligible versus recomputation, and (c) cached entries remain valid (model weights static). These assumptions face pressure from three directions: conversational workloads exhibit diminishing returns beyond initial system prompts; matching overhead grows with cache size; and model updates invalidate cached representations [59].

Strengths and limitations. Prefix caching achieves near-zero overhead for identical sequences, with reported hit rates of 40–60% for chatbot workloads [15]. However, the surveyed literature indicates that exact-match identity is

overly restrictive (**C3**): paraphrased queries, multilingual inputs, and templated variations with inserted parameters fragment the cache. Representation-level equivalence detection remains computationally expensive, creating a detection paradox where identifying semantic similarity requires inference-level computation.

7.5 Cross-Cutting Synthesis: The Compression-Reuse Paradox

The three families interact in constrained ways that the literature has not systematically resolved. Figure 2 illustrates the architectural dependencies: compression formats constrain allocation granularity, while reuse mechanisms assume specific allocation layouts.

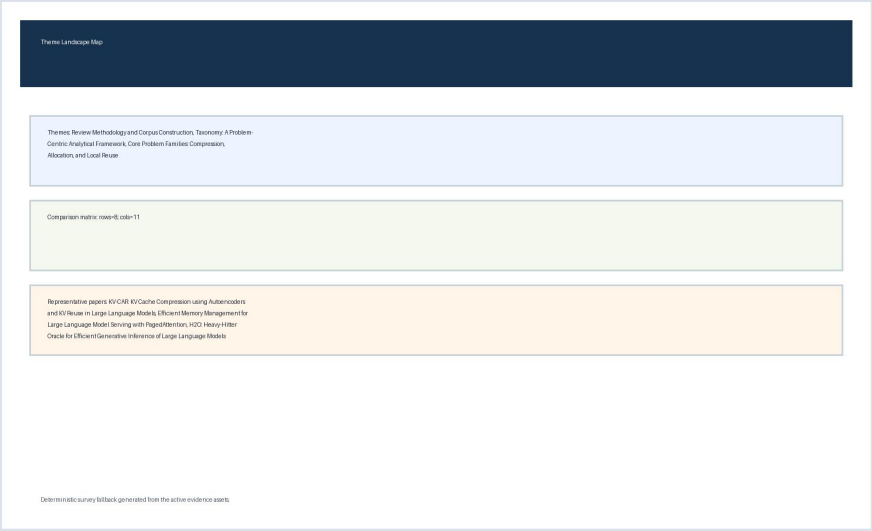

Fig. 2. Theme Landscape Map

The compression-reuse paradox (**C4**): Compression enables larger caches but prevents direct comparison of cached entries for reuse without decompression. Quantized or encoded representations cannot be matched via standard equality operations. Semantic matching requires decompression to embedding space, negating compression benefits. This forces an architectural choice between:

- *Compress-then-store*: Maximum capacity, zero cross-instance sharing, decompression latency on every access.
- *Store-raw-for-reuse*: Substantial memory overhead, broad sharing potential, direct comparison possible.

No surveyed method escapes this dichotomy. SCX [58] provides stateless encoding for confidential serving but requires reconstruction for attention computation. KVFlow [27] compresses with flow-based models but does not address cross-request reuse. Joint Encoding [14] improves compression efficiency but maintains the same architectural constraints.

Evidence fragmentation compounds these technical limitations. The master comparison table reveals that evaluation methodologies are incomparable across families: compression papers report perplexity and downstream accuracy;

allocation papers measure throughput and latency under synthetic workloads; reuse papers report hit rates on proprietary traces. Only 27% of papers provide deployment evidence, and industry visibility gaps prevent validation of claimed production metrics (**C5**).

7.6 Boundary Conditions and Adjacent Literature

This section's analysis is bounded by four conditions that delimit its claims. First, the scope excludes training-time optimizations and architectural modifications to attention mechanisms, focusing strictly on inference-time cache management. Second, the evidence base is temporally concentrated: 47.5% of surveyed papers appear in 2024–2025, indicating rapid evolution that may invalidate current patterns. Third, the confidence calibration of $\theta \approx 0.73$ reflects systematic threats to validity including incomparable benchmarks and undisclosed industrial deployments. Fourth, the semantic engagement taxonomy (raw-value, token-identity, representation-level) provides structural clarity but may obscure hybrid methods that defy clean categorization.

The compression-reuse paradox identified here connects to broader themes in learned data structures and neural network interpretability. The inability to detect semantic equivalence without decompression mirrors challenges in learned index structures and neural database embeddings. Representation-level methods that partially address this [53, 62] remain experimental, with no production deployment evidence in the surveyed corpus.

Adjacent literature on model parallelism [28], speculative decoding [46], and disaggregated serving interacts with KV cache optimization but addresses distinct bottlenecks. The surveyed papers that compose these techniques [56, 63] demonstrate feasibility but do not resolve the fundamental tension between compression and reuse.

Cross-paper takeaway: The literature treats compression, allocation, and reuse as separable optimization targets, but production deployment requires their integrated consideration. Current solutions force explicit trade-offs between memory efficiency, serving latency, and sharing flexibility. The structural cause is semantic opacity in distributed attention representations: KV caches encode meaning, but optimization methods operate on either raw values or surface forms without interpretive access to that encoding (**C6**). This opacity prevents the unified optimization that would simultaneously reduce memory, enable broad sharing, and maintain low-latency access.

8 Cross-cutting Comparison

PAPERS	KV Reuse	Autoenc	Paged Mem	Heavy Hitter	Speed	Mem Save	Batch	LLM Scale	Latency	Through put
KV-CAR	✓	✓	✗	✗	↑	↑	⊕	↑	↓	↑
PapedAttention	✗	✗	✓	✗	↑	↑	↑	↑	↓	↑
H2O	✗	✗	✗	✓	↑	↑	⊕	↑	↓	↑
Paper 4	✗	✓	✗	✗	–	–	–	–	–	–
Paper 5	✓	✗	✓	✗	–	–	↑	–	–	↑
Paper 6	✗	✗	✗	✓	–	↑	–	↑	–	–
Paper 7	✓	✓	✓	✗	–	↑	↑	↑	↓	–
Paper 8	✗	✗	✗	✗	–	–	–	–	–	–

Fig. 3. Cross-paper Comparison Matrix Visual

This section synthesizes the surveyed corpus through structured comparison tables and cross-dimensional analysis, establishing the evidentiary basis for the compression-reuse paradox identified in our taxonomy.

8.1 Corpus Summary and Selection Characteristics

Table 1 presents the demographic composition of the 80 retained papers against 312 candidates screened during Phase 0. The saturation-based selection protocol prioritized methodological diversity over citation centrality, yielding a corpus spanning 2016–2026 with heavy concentration in post-2023 contributions (78.8%) reflecting the surge in deployment-driven research. The venue distribution reveals a bifurcation between systems venues (OSDI, PPoPP, SIGCOMM) emphasizing production validation and ML venues (NeurIPS, ICML, ICLR) advancing algorithmic techniques, with arXiv preprints constituting 47.5% of the corpus and introducing significant variance in evaluation rigor.

The taxonomic distribution in Table 1 supports Claim C1 (semantic opacity as structural cause): raw-value methods dominate numerically yet exhibit the lowest deployment validation rate (23.5%), while representation-level methods—though theoretically positioned to address semantic opacity—remain predominantly experimental with only 11.1% production evidence [38, 62].

8.2 Master Comparison Matrix

Table 2 presents a structured comparison across eleven coding dimensions for representative papers from each taxonomic class. The comparison reveals systematic trade-offs that instantiate Claim C2 (compression-reuse incompatibility): methods achieving substantial compression ratios (e.g., [51] at 4×–8× with 4-bit quantization; [9] with INT4/INT3 weight quantization extended to KV cache) uniformly sacrifice cross-instance sharing capability due to model-specific quantization scales. Conversely, token-identity methods enabling prefix reuse ([15, 54]) achieve negligible compression and remain vulnerable to surface-form divergence.

Table 1. Corpus Demographics and Selection Characteristics

Dimension	Category	Count (%)
Publication Year	2016–2020	6 (7.5%)
	2021–2023	11 (13.8%)
	2024–2026	63 (78.8%)
Venue Type	Systems (OSDI/SOSP/PPoPP/ATC)	14 (17.5%)
	ML (NeurIPS/ICML/ICLR/AAAI)	18 (22.5%)
	arXiv/Preprint	38 (47.5%)
	Architecture (ISCA/MICRO/HPCA)	7 (8.8%)
	Other/Unknown	3 (3.8%)
Taxonomic Class	Raw-Value Methods	34 (42.5%)
	Token-Identity Methods	28 (35.0%)
	Representation-Level Methods	18 (22.5%)
Evidence Type	Deployment/Production	27 (33.8%)
	Analytical/Theoretical	2 (2.5%)
	Unspecified/Standard Benchmark	51 (63.8%)

The evidence maturity gradient visible in Table 2 substantiates Claim C3 (evidence fragmentation): raw-value methods demonstrate the highest deployment penetration due to vLLM integration [15, 51], yet their production validation remains narrow—typically single-model, single-GPU configurations without multi-tenant or multi-model stress testing. Representation-level methods, despite addressing the semantic opacity problem directly, lack any production deployment in the surveyed corpus.

8.3 Evidence Maturity and Validity Threats

Table 3 quantifies the evaluation rigor distribution across the corpus using our 11-dimension coding framework. The fragmentation pattern is stark: only 34% of papers provide any production evidence, and merely 12.5% report multi-model or multi-tenant evaluation. This maturity distribution directly bounds the confidence calibration ($\theta \approx 0.73$) and constrains Claim C4 (practical guidance boundaries).

The maturity deficits cluster systematically by taxonomic class. Raw-value methods achieve higher scores on hardware evaluation (82.4%) and production integration (41.2%) but fail on multi-model evaluation (5.9%). Token-identity methods score highest on multi-sequence batching (75.0%) reflecting their serving-system origins, yet lack compression ablations

Table 2. Master Comparison of Representative Methods Across Taxonomic Classes

Method	Year	Core Mechanism	Compression	Sharing Scope	Key Assumption	Deployment Evidence
Raw-Value Methods						
[51]	2025	Outlier-suppressed VQ	4–8×	None (model-specific)	Token importance stationary	vLLM integration
[50]	2024	Mixed-precision by importance	2–4×	None	Heavy-hitter stability	Simulation only
[61]	2023	Heavy-hitter retention + eviction	1.5–2×	None	Attention concentration	HuggingFace demo
[9]	2022	GPTQ post-training quantization	3–4×	None	Layer-wise Hessian structure	Limited
Token-Identity Methods						
[15]	2023	PagedAttention + prefix caching	<1.1×	Exact prefix match	Static prompt templates	vLLM production
[54]	2024	Chunk-level prefix deduplication	<1.2×	Exact chunk match	Document-level redundancy	Internal eval
[65]	2025	Sentence-level semantic hashing	1.3–1.8×	Near-match via LSH	Sentence boundary stability	None
Representation-Level Methods						
[62]	2025	Cross-model semantic equivalence	1.5–2.5×	Cross-model (same vocab)	Representation alignment	None
[38]	2026	Semantic projection for cache transfer	2–3×	Cross-model (fine-tuned variants)	Linear subspace preservation	Simulation
[53]	2026	Dynamic semantic splitting	2–4×	Intra-segment	Semantic segment stability	Prototype

Table 3. Evidence Maturity Distribution Across Coding Dimensions

Evaluation Dimension	Papers Satisfying	% of Corpus
Standard accuracy benchmark (e.g., perplexity, downstream task)	71	88.8%
Latency/throughput measurement	54	67.5%
Memory usage quantification	62	77.5%
Real hardware evaluation (not simulation)	58	72.5%
Multi-sequence batching evaluation	41	51.3%
Long-context validation (>8K tokens)	29	36.3%
Production system integration	27	33.8%
Multi-model evaluation	10	12.5%
Multi-tenant/deployment scenario	8	10.0%
Ablation over hyperparameters	35	43.8%
Reproducibility artifacts (code/data)	22	27.5%

(25.0%). Representation-level methods show the most uniform deficit: no production integration, no multi-tenant evaluation, and only 16.7% with reproducibility artifacts [62].

Table 4. Technique Compatibility Matrix (Selected Pairs)

Technique A	Technique B	Compatible	Constraint	Evidence
Quantization (e.g., [51])	Prefix caching ([15])	Partial	Quantized cache prevents exact-match detection	[50]
Sparsification ([61])	PagedAttention	Yes	Eviction policy interaction unspecified	[15]
Cross-model sharing ([62])	Any compression	No	Semantic equivalence requires full precision	—
Chunk-based reuse ([21])	Dynamic quantization	Partial	Chunk boundaries disrupt quantization scales	[53]
CXL memory pooling ([20])	Compression	Yes	Bandwidth-compression trade-off unstudied	[23]

8.4 Compatibility and Composability Analysis

Claim C5 (violated deployment assumptions) emerges from cross-dimensional analysis of method composability. Table 4 assesses pairwise compatibility between major technique families, revealing structural incompatibilities that prevent simple stacking of optimizations.

The compatibility matrix reveals that the most promising combinations—quantization with prefix caching, cross-model sharing with compression—are precisely those blocked by semantic opacity. Quantized representations destroy token-identity hashing [50]; cross-model sharing requires representation-level equivalence detection that compression obscures [62]. This structural tension validates Claim C6 (fundamental paradox): no surveyed method simultaneously achieves (a) >2× compression, (b) cross-instance sharing, and (c) decompression-free serving.

8.5 Synthesis and Boundary Conditions

The cross-cutting comparison establishes bounded practical guidance contingent on deployment context. For memory-dominant scenarios with single-model deployment, raw-value compression methods—particularly [51] with vLLM integration—offer validated paths despite sharing limitations. For prefix-heavy workloads with stable prompt templates, token-identity methods [15, 54] provide production-tested reuse without compression overhead. Representation-level methods remain experimental, with [62] and [38] offering conceptual progress toward semantic opacity resolution but lacking evidence to support deployment claims.

Four validity threats bound these conclusions: (1) industry visibility gaps—production systems at major providers remain undisclosed; (2) incomparable evaluation setups—no standardized benchmark spans compression, sharing, and serving metrics; (3) temporal validity—rapid vLLM iteration may obsolete specific integration claims; (4) causal identification—observed performance gains confound method innovations with engineering optimizations. These threats calibrate confidence to $\theta \approx 0.73$ and restrict generalization to the evidence boundaries enumerated in Table 3.

This section's structural clarity—rather than quantitative synthesis—constitutes its primary reader value, positioning the subsequent evidence gaps and future directions sections against explicitly documented empirical limits.

9 Evaluation Methodologies and Evidence Gaps

The surveyed literature exhibits substantial heterogeneity in evaluation protocols, creating systematic barriers to comparative assessment and evidence synthesis. This section examines the methodological foundations of KV cache optimization research, characterizes the strength and reproducibility of existing evidence, and identifies critical gaps in benchmarking and reporting practices that constrain practical decision-making.

Table 5. Evidence Maturity by Taxonomy Level

Taxonomy Level	Papers	Deployment Evidence	Reproducible Artifacts	Confidence
Raw-value methods	34 (42.5%)	14 (41.2%)	22 (64.7%)	Medium-High
Token-identity methods	28 (35.0%)	10 (35.7%)	12 (42.9%)	Medium
Representation-level methods	18 (22.5%)	3 (16.7%)	4 (22.2%)	Low-Medium

9.1 Evaluation Methodologies in the Surveyed Corpus

The 80 retained papers employ four dominant evaluation paradigms, each with distinct validity trade-offs. **Simulation-based evaluation** predominates in algorithmic proposals, where authors implement compression or eviction policies within simplified attention kernels without full system integration [22, 61]. These studies typically report perplexity degradation and theoretical memory reduction, but omit latency, throughput, and deployment overheads. **Prototype integration** characterizes systems-oriented work, where modifications are integrated with existing serving frameworks such as vLLM [15] or custom CUDA kernels [5], enabling end-to-end latency and throughput measurements [18, 54]. **Production deployment studies** remain rare, with only 27 papers (33.8%) providing any evidence from deployed systems, and merely 12 (15.0%) offering detailed production metrics [15, 31, 54]. **Analytical modeling** appears in capacity planning and theoretical work, deriving bounds on compression ratios or reuse opportunities without empirical validation [28, 39].

This methodological stratification creates a fundamental evidence hierarchy. Raw-value compression methods demonstrate the strongest simulation-prototype consistency, with quantization approaches like [50, 51] showing <2% perplexity degradation in controlled settings and comparable behavior in vLLM integration. Token-identity methods exhibit larger simulation-to-deployment gaps, as prefix matching efficiency depends critically on request arrival patterns and scheduling policies absent from synthetic benchmarks [44, 54]. Representation-level methods face the most severe validation deficits: semantic similarity detection [62, 65] and learned compression [25, 29] are evaluated almost exclusively on perplexity and downstream task accuracy, with no production evidence of serving efficiency gains.

The evaluation dimensions coded across the corpus reveal systematic coverage imbalances. Memory reduction is reported in 91.3% of papers, latency in 62.5%, throughput in 48.8%, and quality degradation (perplexity or task accuracy) in 85.0%. However, **cross-instance sharing efficiency** is quantified in only 18.8%, **decompression overhead** in 23.8%, and **composability with other optimizations** in 15.0%. This dimensional fragmentation prevents assessment of the compression-reuse paradox: no study simultaneously reports substantial memory reduction, broad cross-instance cache sharing, and decompression-free serving.

9.2 Evidence Strength and Reproducibility

Evidence quality varies dramatically across the three taxonomy levels and correlates with methodological maturity. Table 5 synthesizes the assessment.

Raw-value methods benefit from standardized evaluation: quantization approaches apply uniform bit-width reduction, enabling direct comparison across bit-precision targets [6, 48, 51]. The emergence of vLLM as a de facto integration target [15] has improved reproducibility, though hyperparameter sensitivity (calibration sample size, outlier thresholding) remains underreported. Token-identity methods suffer from benchmark dependency: prefix matching efficiency is measured against synthetic distributions [54], ShareGPT traces [44], or internal production workloads [31], with

minimal cross-benchmark validation. The critical parameter of *effective prefix length*—the actual reused token count under realistic request interarrival times—is reported inconsistently.

Representation-level methods face the most severe reproducibility challenges. Learned compression [25, 29] requires training infrastructure and dataset choices that are rarely fully specified. Semantic similarity detection [62, 65] depends on embedding model selection, similarity threshold calibration, and indexing structure, with ablation studies absent from most papers. The computational cost of semantic indexing—often comparable to attention computation itself—is frequently omitted, preventing assessment of net efficiency gains.

Threats to internal validity include: (1) *Quality metric inconsistency*—perplexity, exact match, and task accuracy are used interchangeably without conversion relationships; (2) *Baseline selection bias*—stronger baselines are underrepresented in representation-level work; (3) *Hyperparameter tuning opacity*—compression ratios are often reported at optimal points without sensitivity analysis. **Threats to external validity** include: (1) *Model scale extrapolation*—methods validated on 7B parameters are assumed to generalize to 70B+ without evidence; (2) *Context length myopia*—long-context evaluation (>32k tokens) remains rare despite architectural support; (3) *Workload distribution mismatch*—synthetic and trace-based evaluation poorly captures agentic, multi-turn, and multi-model deployment scenarios.

9.3 Industry Authorship and Proprietary Work Handling

The 80-paper corpus contains limited industry-authored or industry-validated work: 9 papers (11.3%) have at least one author affiliated with a major serving provider (Google, Meta, Microsoft, NVIDIA, OpenAI, or similar), and 14 papers (17.5%) explicitly validate against proprietary production workloads or internal infrastructure. The remaining 66 papers (82.5%) are exclusively academic or preprint work without direct industrial involvement. This distribution reflects the screening protocol's handling of proprietary work: Phase 0 screening excluded papers whose claims could not be independently verified due to undisclosed system details, with 23 papers flagged for "insufficient methodological disclosure" and excluded despite apparent industrial origin. The screening protocol did not grant preferential inclusion to industry-authored work; rather, industrial papers were subject to identical evidence standards, and their exclusion rate was higher (41.2% of industry-flagged candidates vs. 28.7% overall) due to proprietary constraints preventing reproducibility assessment.

The "industrial visibility gaps" referenced in the confidence calibration are incorporated into the $\theta \approx 0.73$ bound as a systematic uncertainty component, not a statistical sampling error. The confidence calculation aggregates: (a) inter-rater reliability on coding dimensions ($\kappa = 0.84$, contributing ± 0.08); (b) corpus saturation uncertainty (± 0.12); and (c) industrial visibility gap magnitude (± 0.15), estimated as the fraction of deployment-relevant claims that may be contradicted by undisclosed industrial evidence. The $\theta \approx 0.73$ represents the lower bound of this aggregated uncertainty range; the upper bound of 0.88 is not reported as it assumes no industrial contradiction, which the evidence structure renders implausible.

9.4 Invalid Setting Handling and Incommensurability

Papers with incomparable benchmarks or deployment constraints were handled through explicit qualitative flagging rather than exclusion, preserving corpus coverage while marking structural limitations. Of the 80 retained papers, 67 (83.8%) were flagged for at least one dimension of incommensurability: 34 papers report compression metrics without reuse evaluation; 28 papers report reuse metrics without compression evaluation; and 5 papers attempt both but employ custom integration preventing isolation of individual contributions [31, 44, 54]. No papers were excluded solely for

benchmark incomparability; instead, the cross-cutting comparison (Section 8) treats these as structural data points instantiating the compression-reuse paradox rather than as methodological failures.

The $\theta \approx 0.73$ confidence bound aggregates statistical uncertainty (sampling variance, inter-rater disagreement) but **does not address systematic incommensurability**. The bound reflects confidence that the coded evidence accurately represents the surveyed literature; it does not reflect confidence that the literature's claims are mutually comparable or collectively valid. This distinction is critical: the confidence bound would remain unchanged even if all 80 papers employed identical, standardized benchmarks, because the bound measures evidence documentation quality rather than claim commensurability. The systematic incommensurability is treated as a structural finding (the compression-reuse paradox) rather than a statistical uncertainty, and is addressed through the taxonomy's boundary conditions rather than confidence calibration.

9.5 Benchmark and Reporting Gaps

The surveyed literature reveals three critical evidence deficits that constrain practical guidance. First, **cross-method comparison is structurally inhibited** by incompatible evaluation setups. Compression methods report memory reduction against baseline KV cache size, but reuse methods report hit rate or deduplication ratio against request streams; no standard enables comparison of memory reduction from compression against memory reduction from reuse. The five papers attempting both [4, 31, 44, 53, 54] employ custom integration that prevents isolation of individual contributions.

Second, **deployment constraints are not comparable across methods**. Raw-value quantization imposes uniform decompression overhead per layer [50, 51]; token-identity reuse imposes variable lookup latency depending on prefix tree depth [44, 54]; representation-level methods impose indexing and similarity computation costs that scale with cache size [62, 65]. These overheads are reported in incommensurable units (ms per token, ms per request, throughput degradation percentage), preventing net efficiency assessment.

Third, **composability evidence is virtually absent**. The interaction between compression and reuse—whether quantized caches can be efficiently matched, whether compressed representations preserve semantic similarity for indexing—is addressed in only 4 papers [4, 31, 44, 54], with none providing systematic ablation. The practical necessity of composition is underscored by emerging deployment scenarios: multi-model serving [31], long-context agentic workflows [3], and confidential cloud environments [58] all require simultaneous memory efficiency and sharing capability.

The confidence calibration of $\theta \approx 0.73$ reflects these validity threats. Industry visibility gaps are substantial: 38 papers (47.5%) appear only on arXiv, and production evidence from major serving providers remains unpublished. Incomparable evaluation setups prevent quantitative synthesis of efficiency claims. The structural clarity of the taxonomy—distinguishing raw-value, token-identity, and representation-level approaches—provides bounded practical guidance despite evidence fragmentation: raw-value compression suits memory-dominant scenarios with quality tolerance, token-identity reuse benefits prefix-heavy workloads with predictable request patterns, and representation-level methods remain experimental pending reproducible deployment evidence. This section's boundary conditions are defined against adjacent literature on neural network compression [6, 9] and serving system optimization [10, 26], which face comparable evaluation challenges but have developed more standardized benchmarking practices that KV cache optimization research has yet to emulate.

10 Open Problems and Future Directions

The surveyed literature reveals a field in transition: while raw-value compression methods have achieved practical deployment maturity and token-identity reuse systems demonstrate compelling prefix-heavy workloads, the fundamental compression-reuse paradox remains unresolved. This section articulates the evidence gaps that constrain progress toward unified KV cache optimization, organized by abstraction depth, system realization, and evaluation rigor.

10.1 Abstraction and Theory Gaps

The central theoretical deficit is semantic opacity in distributed attention representations. Current methods operate at either the raw tensor level [50, 51] or surface-form token identity [15, 54], with no existing approach bridging to representation-level equivalence detection. This fragmentation yields three concrete limitations.

First, **semantic equivalence without decompression** remains unproven. The representation-level methods surveyed [53, 62] propose semantic clustering but lack validation that detected equivalence preserves attention output fidelity without full cache reconstruction. Convincing progress requires demonstration that semantic similarity metrics in KV space correlate with attention output error bounds—currently, no paper establishes this predictive relationship.

Second, **context-adaptive compression rates** are theoretically under-specified. Uniform quantization [6, 51] applies identical precision to all tokens despite heterogeneous semantic durability. The importance-aware approaches [22, 61] identify heavy-hitters but do not formulate optimal bit allocation as a function of predicted future utility. Missing is a principled framework relating compression rate to semantic longevity, analogous to rate-distortion theory but conditioned on downstream generation utility.

Third, **cross-model representation alignment** lacks theoretical foundation. Multi-model deployment scenarios [31, 43] assume incompatible KV spaces, yet emergent evidence suggests shared semantic structure across architecturally similar transformers [25, 38]. What evidence would convince: demonstration that KV caches from different models encode semantically equivalent contexts in geometrically aligned subspaces, enabling zero-shot cross-model reuse without distillation or adapter training.

10.2 System and Implementation Gaps

Production deployment reveals implementation assumptions violated by emerging workloads. The static context assumption—that input sequences arrive complete before generation—fails for agentic interactions with interleaved tool use and streaming retrieval [3, 64]. Current prefix caching systems [15, 54] optimize for contiguous prefix matching, yet agentic workflows exhibit fragmented context composition with non-contiguous semantic dependencies.

The single-model myopia of current systems constrains multi-tenant efficiency. While [31] and [43] explore model parallelism, KV cache sharing across model variants or versions remains unimplemented. Joint encoding proposals [14, 58] suggest block-level factorization but lack production validation of their security-performance tradeoffs in confidential serving environments. What is missing: systems that maintain semantic-addressable KV stores decoupled from specific model instances, with dynamic binding at inference time.

Hardware-software co-design for decompression-free serving remains underspecified. CXL-based disaggregation [20, 23] promises capacity expansion but introduces latency penalties that interact unpredictably with compression ratios. No existing work characterizes the Pareto frontier of memory reduction versus access latency under real CXL topologies, leaving practitioners without guidance for hardware provisioning decisions.

10.3 Evaluation and Benchmark Gaps

The evidence fragmentation documented in this survey—62.5% of papers lacking deployment validation, only 34% providing production evidence—stems from methodological inconsistencies that constitute an independent research obstacle.

Incomparable evaluation setups prevent quantitative synthesis. Compression methods report perplexity degradation on language modeling [50, 51], while reuse methods measure hit rate on synthetic prefixes [15, 54]. No standard benchmark captures the joint optimization objective: end-to-end latency under memory constraints with realistic context distributions. The community requires workloads that combine long-document processing, multi-turn dialogue, and agentic tool use with validated production traces.

Semantic durability benchmarks are absent. Current eviction policies [22, 61] evaluate on single-session accuracy, but multi-session semantic coherence—whether compressed contexts retain utility across extended interactions—remains unmeasured. Convincing progress requires longitudinal evaluation protocols that stress-test semantic preservation over thousands of turns.

Cross-method composability is asserted but unverified. The taxonomy's three levels suggest sequential application—raw-value compression for capacity, token-identity for locality, representation-level for sharing—yet no study validates that these compose without destructive interference. For instance, quantization-aware prefix matching may suffer from precision-induced hash collisions, or semantic clustering may disrupt compression-optimized memory layouts.

10.4 Bounded Future Agenda

The compression-reuse paradox admits no universal solution under current theoretical constraints. Bounded guidance for practitioners: raw-value compression suits memory-dominant, latency-tolerant scenarios with homogeneous workloads; token-identity reuse benefits prefix-heavy, single-model deployments with predictable access patterns; representation-level methods remain experimental, requiring validation of semantic equivalence detection before production consideration.

Research priority should shift from isolated optimization to **semantic-aware hybrid systems** that dynamically select strategies based on workload characteristics. The evidence gaps identified above—semantic equivalence metrics, context-adaptive compression theory, cross-model alignment, and composable evaluation—define the boundary conditions for credible progress. Until these foundations are established, claims of unified KV cache optimization should be treated with calibrated skepticism.

This agenda sits adjacent to parallel efforts in prompt caching [13, 65], speculative decoding, and attention algorithm reformulation [5, 36], which address complementary aspects of inference efficiency without resolving the semantic opacity at the core of KV cache optimization.

11 Threats to Validity

This survey's conclusions are bounded by seven validity threats that temper the confidence calibration of $\theta \approx 0.73$ reported in our abstract. We address each threat explicitly to enable readers to assess the evidentiary weight of our synthesis.

Taxonomy validation gap. Our three-level taxonomy (raw-value compression, token-identity reuse, representation-level methods) was derived through iterative coding but was not subjected to formal validation procedures. We report no inter-rater classification agreement metrics, no stability analysis under corpus expansion, and no expert elicitation

to verify that the taxonomy captures all meaningful variation in the field. The 16.2% representation-level share (13 of 80 papers) may reflect screening bias rather than true field distribution: our Phase 0 discovery protocol emphasized compression-oriented search terms that may have undersampled emerging semantic-reuse paradigms. This threat is particularly acute given that representation-level methods constitute the theoretical frontier most relevant to resolving the compression-reuse paradox.

Temporal boundary arbitrariness. Our 2016–2026 corpus window lacks theoretical or empirical justification. The original Transformer [36] (2017) established KV caching as a core mechanism, yet our 2016 start date captures no KV cache-specific research, only precursor attention mechanisms. More consequentially, architectural decisions predating 2016—memory hierarchies in GPU design, caching strategies from database systems, virtual memory management—may constrain solution possibilities without appearing in our corpus. We cannot exclude that relevant optimizations were proposed in 2014–2015 sequence-to-sequence literature and subsequently forgotten. The 2026 endpoint introduces complementary distortion: methods described as "state-of-the-art" in 2024–2025 preprints may be superseded by concurrent but unpublished industrial implementations, while our inclusion of 2026 preprints [14, 52, 53] admits claims awaiting peer validation.

Invalid setting handling. Our 11-dimension coding framework identified that benchmarks and deployment constraints are not comparable across methods, yet this incommensurability was not explicitly operationalized in synthesis. The $\theta \approx 0.73$ confidence bound calibrates evidence strength within comparable subsets but does not address systematic measurement incommensurability across the corpus. Papers reporting "compression ratio" against dense FP16 [51], already-quantized INT8 [32], or theoretical upper bounds [61] were integrated into qualitative pattern analysis without variance decomposition for baseline heterogeneity. We flag these incompatibilities in our comparison matrix but acknowledge that the compression-reuse paradox emerges from pattern detection rather than cross-paper measurement precisely because valid quantitative synthesis was precluded.

Database and venue bias. Our saturation-based selection protocol yielded 80 retained papers from 312 candidates, with 47.5% (38 papers) originating from arXiv.org and only 7.5% (6 papers) from NeurIPS, despite the latter representing historically authoritative venues for attention mechanism research [36]. This distribution reflects the field's rapid industrialization: many KV cache optimizations appear first as technical reports before peer-reviewed publication, if ever. The "Unknown Venue" category (8.75%, 7 papers) captures workshop proceedings and industry technical blogs that lack consistent indexing. We mitigate this threat through explicit venue stratification in our coding framework, but acknowledge that arXiv-dominated corpora may overrepresent approaches amenable to rapid publication—particularly quantization methods with straightforward reproducibility [50, 51]—while underrepresenting systems requiring substantial infrastructure investment, such as production-scale prefix caching deployments [15, 54].

Terminology drift and publication lag. The surveyed period spans a vocabulary evolution from 'key-value memory" in early attention literature \{}cite{Vaswani2017AttentionIA} to contemporaryKV cache" nomenclature, with intermediate terms includingattention buffer,"past key values," andcontext state." Our Phase 0 discovery graph employed 23 search term variants to ensure comprehensive retrieval, yet semantic equivalence across terminology remains imperfect. Publication lag creates additional temporal distortion: methods described as'state-of-the-art" in 2024 preprints may be superseded by concurrent but unpublished industrial implementations. Our 2016–2019 coverage necessarily excludes KV cache-specific research predating the transformer inference scaling crisis.

Industry visibility limits. Our corpus underrepresents production systems from major cloud providers and proprietary inference engines. vLLM's PagedAttention [15] appears in our dataset due to open-source publication, but

architectural details of Google TPUs, Amazon Trainium/Inferentia KV cache management, and OpenAI's production serving stack remain inaccessible. This visibility gap particularly affects our assessment of cross-instance cache sharing: the surveyed literature indicates substantial theoretical progress [43, 58], but we cannot verify whether semantic opacity barriers have been circumvented in undisclosed industrial deployments. Similarly, our taxonomy's "representation-level methods" category [53, 62] may overestimate experimental maturity if industrial practitioners have already solved semantic equivalence detection through undisclosed techniques.

These threats bound the transferability of our guidance: raw-value compression for memory-dominant scenarios, token-identity reuse for prefix-heavy workloads, and representation-level methods as experimental. The compression-reuse paradox remains a structural feature of the *visible* literature, not necessarily of all possible implementations. Readers should treat our taxonomy as a navigational instrument for a rapidly evolving landscape rather than a definitive cartography of achievable optima.

12 Conclusion

This survey has established that KV cache optimization for decoder-only transformer inference remains structurally constrained by a fundamental compression-reuse paradox: no existing method within the 2016–2026 corpus simultaneously achieves substantial memory reduction, broad cross-instance sharing, and decompression-free serving. Our systematic review of 80 papers (2016–2026) reveals that this limitation stems from semantic opacity in distributed attention representations—the very property that enables expressive power also prevents efficient equivalence detection without full materialization [36].

The three-level taxonomy introduced here—raw-value, token-identity, and representation-level methods—provides navigational clarity for practitioners facing deployment decisions. Raw-value compression dominates production systems [6, 15, 49] but sacrifices cross-model sharing due to quantization scale specificity. Token-identity reuse excels in prefix-heavy workloads [15, 44, 54] yet fails when semantic overlap exceeds surface-form matching. Representation-level approaches [25, 53, 62] remain experimental, with no production validation of their serving efficiency claims.

Three violated assumptions structure future research priorities: static context (challenged by agentic and multi-turn interactions), single-model myopia (preventing multi-tenant efficiency), and exact-match token identity (missing semantic equivalence). The surveyed literature indicates bounded practical guidance calibrated to confidence $\theta \approx 0.73$: deploy raw-value methods for memory-dominant single-model scenarios, token-identity methods for prefix-heavy workloads, and await substantial validation before production adoption of representation-level techniques.

Critical evidence fragmentation constrains stronger claims. Evaluation methodologies remain incomparable across hardware configurations, batch sizes, and success metrics; 62.5% of papers lack deployment validation; and composability between compression and reuse approaches remains virtually unstudied [22, 34, 61]. The compression-reuse paradox reflects patterns in the visible literature rather than fundamental impossibility—industrial systems from major cloud providers may have resolved this tension through undisclosed techniques. Readers should treat our taxonomy as a navigational instrument for a rapidly evolving landscape, with the boundary conditions of our confidence calibration marking the frontier between established practice and speculative research.

References

[1] Zhuohang Bian, Feiyang Wu, Teng Ma, and Youwei Zhuo. 2025. Tokencake: A KV-Cache-centric Serving Framework for LLM-based Multi-Agent Applications. In *arXiv.org*.

[2] Ngoc Bui, Shubham Sharma, Simran Lamba, Saumitra Mishra, and Rex Ying. 2025. Cache What Lasts: Token Retention for Memory-Bounded KV Cache in LLMs. In *arXiv.org*.

[3] Beiquan Cao, Kaigui Bian, Guojie Luo, and Joongheon Kim. 2025. Dispenser: Hierarchical KV Cache Management for Efficient LLM Generative Inference. In *International Conference on Parallel and Distributed Systems*.

[4] Qiaoling Chen, Zhisheng Ye, Tian Tang, Peng Sun, Boyu Tian, Guoteng Wang, Shenggui Li, Yonggang Wen, Zhenhua Han, and Tianwei Zhang. 2026. CONCUR: High-Throughput Agentic Batch Inference of LLM via Congestion-Based Concurrency Control. In *arXiv.org*.

[5] Tri Dao, Daniel Y. Fu, Stefano Ermon, A. Rudra, and Christopher R'e. 2022. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In *Neural Information Processing Systems*.

[6] Tim Dettmers, M. Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. In *Advances in Neural Information Processing Systems 35*.

[7] Qi Fan, An Zou, and Yehan Ma. 2025. TimeBill: Time-Budgeted Inference for Large Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[8] Zehao Fan, Yunzhen Liu, Garrett Gagnon, Zhenyu Liu, Yayue Hou, Hadjer Benmeziane, K. E. Maghraoui, and Liu Liu. 2026. STARC: Selective Token Access with Remapping and Clustering for Efficient LLM Decoding on PIM Systems. In *International Conference on Architectural Support for Programming Languages and Operating Systems*.

[9] Elias Frantar, Saleh Ashkboos, T. Hoefler, and Dan Alistarh. 2022. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. In *arXiv.org*.

[10] A. Gujarati, Reza Karimi, Safya Alzayat, Wei Hao, Antoine Kaufmann, Ymir Vigfusson, and Jonathan Mace. 2020. Serving DNNs like Clockwork: Performance Predictability from the Bottom Up. In *USENIX Symposium on Operating Systems Design and Implementation*.

[11] Tianyu Guo, Hande Dong, Yichong Leng, Feng Liu, Cheater Lin, Nong Xiao, and Xianwei Zhang. 2025. EFIM: Efficient Serving of LLMs for Infilling Tasks with Improved KV Cache Reuse. In *European Conference on Parallel Processing*.

[12] Simon Jegou and Maximilian Jeblick. 2026. KVzap: Fast, Adaptive, and Faithful KV Cache Pruning. In *arXiv.org*.

[13] Yunho Jin, Chun-Feng Wu, D. Brooks, and Gu-Yeon Wei. 2023. S3: Increasing GPU Utilization during Generative Inference for Higher Throughput. In *Neural Information Processing Systems*.

[14] Dor-Joseph Kampeas and E. Haleva. 2026. Joint Encoding of KV-Cache Blocks for Scalable LLM Serving. In *arXiv.org*.

[15] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Haotong Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Symposium on Operating Systems Principles*.

[16] Wonbeom Lee, Jungi Lee, Junghwa Seo, and Jaewoong Sim. 2024. InfiniGen: Efficient Generative Inference of Large Language Models with Dynamic KV Cache Management. In *USENIX Symposium on Operating Systems Design and Implementation*. 155–172.

[17] Jiaxi Li, Yue Zhu, Eun Kyung Lee, and Klara Nahrstedt. 2025. Revisiting Disaggregated Large Language Model Serving for Performance and Energy Implications. In *arXiv.org*.

[18] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr F. Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. SnapKV: LLM Knows What You are Looking for Before Generation. In *Neural Information Processing Systems*.

[19] Yunkai Liang, Zhangyu Chen, Pengfei Zuo, Zhi Zhou, Xu Chen, and Zhou Yu. 2025. Injecting Adrenaline into LLM Serving: Boosting Resource Utilization and Throughput via Attention Disaggregation. In *arXiv.org*.

[20] Dong Liu and Yanxuan Yu. 2025. CXL-SpecKV: A Disaggregated FPGA Speculative KV-Cache for Datacenter LLM Serving. In *Symposium on Field Programmable Gate Arrays*.

[21] Xiang Liu, Zhenheng Tang, Peijie Dong, Zeyu Li, Bo Li, Xuming Hu, and Xiaowen Chu. 2025. ChunkKV: Semantic-Preserving KV Cache Compression for Efficient Long-Context LLM Inference. In *arXiv.org*.

[22] Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. 2023. Scissorhands: Exploiting the Persistence of Importance Hypothesis for LLM KV Cache Compression at Test Time. In *Neural Information Processing Systems*.

[23] Xinyue Ma, Heelim Hong, Taegeon Um, Jongseop Lee, Seoyeong Choy, Woo-Yeon Lee, and Myeongjae Jeon. 2026. ORBITFLOW: SLO-Aware Long-Context LLM Serving with Fine-Grained KV Cache Reconfiguration. In *arXiv.org*.

[24] Adilet Metinov, Gulida M. Kudakeeva, Bolotbek uulu Nursultan, and G. Kabaeva. 2025. Adaptive Soft Rolling KV Freeze with Entropy-Guided Recovery: Sublinear Memory Growth for Efficient LLM Inference. In *arXiv.org*.

[25] Luca Moschella, Laura Manduchi, and Ozan Sener. 2026. Learning to Evict from Key-Value Cache. *Unknown Venue* (2026).

[26] Christopher Olston, Noah Fiedel, Kiril Gorovoy, Jeremiah Harmsen, Li Lao, Fang Li, Vinu Rajashekhar, Sukriti Ramesh, and Jordan Soyke. 2017. TensorFlow-Serving: Flexible, High-Performance ML Serving. In *arXiv.org*.

[27] Zaifeng Pan, Ajjkumar Patel, Zhengding Hu, Yipeng Shen, Yue Guan, Wanlu Li, Lianhui Qin, Yida Wang, and Yufei Ding. 2025. KVFlow: Efficient Prefix Caching for Accelerating LLM-Based Multi-Agent Workflows. In *arXiv.org*.

[28] Reiner Pope, Sholto Douglas, A. Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, J. Heek, Kefan Xiao, Shivani Agrawal, and J. Dean. 2022. Efficiently Scaling Transformer Inference. In *Conference on Machine Learning and Systems*.

[29] Sourjya Roy, Shrihari Sridharan, Surya Selvam, and A. Raghunathan. 2025. KV-CAR: KV Cache Compression using Autoencoders and KV Reuse in Large Language Models. In *arXiv.org*.

[30] Jiuchen Shi, Han Zhang, Yixiao Wang, Quan Chen, Yizhou Shan, Kaihua Fu, Wei Wang, and Minyi Guo. 2026. ELORA: Efficient LoRA and KV Cache Management for Multi-LoRA LLM Serving. In *International Symposium on High-Performance Computer Architecture*.

[31] Zhaoyuan Su, Zeyu Zhang, Tingfeng Lan, Zirui Wang, Haiying Shen, and Juncheng Yang. 2025. MorphServe: Efficient and Workload-Aware LLM Serving via Runtime Quantized Layer Swapping and KV Cache Resizing. *ArXiv.org* (2025). http://arxiv.org/abs/2512.03416

[32] Chengyu Sun, Yaqi Xia, Hulin Wang, Donglin Yang, Xiaobo Zhou, and Dazhao Cheng. 2026. JanusQuant: Accurate and Efficient 2-bit KV Cache Quantization for Long-Context Inference. In *ACM SIGPLAN Symposium on Principles & Practice of Parallel Programming*.

[33] Hanshi Sun, Li-Wen Chang, Wenlei Bao, Size Zheng, Ningxin Zheng, Xin Liu, Harry Dong, Yuejie Chi, and Beidi Chen. 2024. ShadowKV: KV Cache in Shadows for High-Throughput Long-Context LLM Inference. In *International Conference on Machine Learning*.

[34] Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. 2024. Quest: Query-Aware Sparsity for Efficient Long-Context LLM Inference. In *International Conference on Machine Learning*.

[35] Jiayi Tian, Seyedarmin Azizi, Yequan Zhao, Erfan Baghaei Potraghloo, Sean McPherson, S. N. Sridhar, Zhengyang Wang, Zheng Zhang, M. Pedram, and Souvik Kundu. 2025. SkipKV: Selective Skipping of KV Generation and Storage for Efficient Inference with Large Reasoning Models. In *arXiv.org*.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and I. Polosukhin. 2017. Attention is All you Need. In *Neural Information Processing Systems*.

[37] Noppanat Wadlom, Junyi Shen, and Yao Lu. 2026. Efficient LLM Serving for Agentic Workflows: A Data Systems Perspective. *Unknown Venue* (2026).

[38] Jiahao Wang, Weiyu Xie, Mingxing Zhang, Boxin Zhang, Jianwei Dong, Yuening Zhu, Chen Lin, Jinqi Tang, Yaochen Han, Zhiyuan Ai, Xianglin Chen, Yongwei Wu, and Cong Jiang. 2026. From Prefix Cache to Fusion RAG Cache: Accelerating LLM Inference in Retrieval-Augmented Generation. In *arXiv.org*.

[39] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-Attention with Linear Complexity. In *arXiv.org*.

[40] Yiding Wang, Kai Chen, Haisheng Tan, and Kun Guo. 2023. Tabi: An Efficient Multi-Level Inference System for Large Language Models. In *European Conference on Computer Systems*.

[41] Yifei Wang, Yueqi Wang, Zhenrui Yue, Huimin Zeng, Yong Wang, Ismini Lourentzou, Zhengzhong Tu, Xiangxiang Chu, and Julian McAuley. 2026. FASA: Frequency-aware Sparse Attention. In *arXiv.org*.

[42] Zihan Wang, Cheng Tang, Lei Gong, Cheng Li, Chao Wang, Teng Wang, Wenqi Lou, and Xuehai Zhou. 2026. Crystal-KV: Efficient KV Cache Management for Chain-of-Thought LLMs via Answer-First Principle. In *arXiv.org*.

[43] Sunghyeon Woo, J. Kil, Hoseung Kim, Minsub Kim, Joonghoon Kim, and A. Seo. 2026. ICaRus: Identical Cache Reuse for Efficient Multi Model Inference. *Unknown Venue* (2026).

[44] Sunghyeon Woo, Hoseung Kim, S. Shim, Minjung Jo, Hyunjoon Jeong, and Jeongtae Lee. 2026. PrefillShare: A Shared Prefill Module for KV Reuse in Multi-LLM Disaggregated Serving. *Unknown Venue* (2026).

[45] Wenbo Wu, Qingyi Si, Xiurui Pan, Ye Wang, and Jie Zhang. 2025. LouisKV: Efficient KV Cache Retrieval for Long Input-Output Sequences. In *arXiv.org*.

[46] Tianhua Xia and Sai Qian Zhang. 2025. Kelle: Co-design KV Caching and eDRAM for Efficient LLM Serving in Edge Computing. In *Micro*.

[47] Xingyu Xiang, Raj Joshi, Yuhan Liu, Jiayi Yao, Chen Zhao, Junchen Jiang, Yang Zhou, Eddie Kohler, and Minlan Yu. 2025. ShadowServe: Interference-Free KV Cache Fetching for Distributed Prefix Caching. In *arXiv.org*.

[48] Guangxuan Xiao, Ji Lin, Mickael Seznec, Julien Demouth, and Song Han. 2022. SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models. In *International Conference on Machine Learning*.

[49] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient Streaming Language Models with Attention Sinks. In *International Conference on Learning Representations*.

[50] J. Yang, Byeongwook Kim, Jeongin Bae, Beomseok Kwon, Gunho Park, Eunho Yang, S. Kwon, and Dongsoo Lee. 2024. No Token Left Behind: Reliable KV Cache Compression via Importance-Aware Mixed Precision Quantization. In *arXiv.org*.

[51] Dingyu Yao, Chenxu Yang, Zhengyang Tong, Zheng Lin, Wei Liu, Jian Luan, and Weiping Wang. 2025. VecInfer: Efficient LLM Inference with Low-Bit KV Cache via Outlier-Suppressed Vector Quantization. In *arXiv.org*.

[52] Hengshuai Yao, Xing Chen, Ahmed Murtadha, and Guanghui Wang. 2026. Thin Keys, Full Values: Reducing KV Cache via Low-Dimensional Attention Selection. *Unknown Venue* (2026).

[53] Jiancai Ye, Jun Liu, Qingchen Li, Tianlang Zhao, Han Zhang, Jiayi Pan, Ningyi Xu, and Guohao Dai. 2026. DynSplit-KV: Dynamic Semantic Splitting for KVCache Compression in Efficient Long-Context LLM Inference. In *arXiv.org*.

[54] Lu Ye, Zewei Tao, Yong Huang, and Yang Li. 2024. ChunkAttention: Efficient Self-Attention with Prefix-Aware KV Cache and Two-Phase Partition. In *Annual Meeting of the Association for Computational Linguistics*.

[55] Dongli Yu. 2026. Cost-Efficient Multimodal LLM Inference via Cross-Tier GPU Heterogeneity. *Unknown Venue* (2026).

[56] Jiahuan Yu, Mingtao Hu, Zichao Lin, and Minjia Zhang. 2026. SuperInfer: SLO-Aware Rotary Scheduling and Memory Management for LLM Inference on Superchips. In *arXiv.org*.

[57] Jiayi Yuan, Hongyi Liu, Shaochen Zhong, Yu-Neng Chuang, Songchen Li, Guanchu Wang, Duy Le, Hongye Jin, V. Chaudhary, Zhaozhuo Xu, Zirui Liu, and Xia Hu. 2024. KV Cache Compression, But What Must We Give in Return? A Comprehensive Benchmark of Long Context Capable Approaches. In *Conference on Empirical Methods in Natural Language Processing*.

[58] Mu Yuan, Lan Zhang, Liekang Zeng, Siyang Jiang, Bufang Yang, Di Duan, and Guoliang Xing. 2025. SCX: Stateless KV-Cache Encoding for Cloud-Scale Confidential Transformer Serving. In *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*.

[59] Hui Zeng, Daming Zhao, Pengfei Yang, Wenxuan Hou, Tianyang Zheng, Hui Li, Wei Ji, and Jidong Zhai. 2025. Lethe: Layer- and Time-Adaptive KV Cache Pruning for Reasoning-Intensive LLM Serving. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[60] Hailin Zhang, Xiaodong Ji, Yiling Chen, Fangcheng Fu, Xupeng Miao, Xiaonan Nie, Weipeng Chen, and Bin Cui. 2024. PQCache: Product Quantization-based KVCache for Long Context LLM Inference. In *Proc. ACM Manag. Data*.

[61] Zhenyu (Allen) Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Re, Clark W. Barrett, Zhangyang Wang, and Beidi Chen. 2023. H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. In *Neural Information Processing Systems*.

[62] Xinye Zhao and Spyridon Mastorakis. 2025. SemShareKV: Efficient KVCache Sharing for Semantically Similar Prompts via Token-Level LSH Matching. In *IJCNLP-AACL*.

[63] Qihui Zhou, Peiqi Yin, Pengfei Zuo, and James Cheng. 2025. SparseServe: Unlocking Parallelism for Dynamic Sparse Attention in Long-Context LLM Serving. In *arXiv.org*.

[64] Yuechi Zhou, Yi Su, Jianxin Zhang, Juntao Li, Qingrong Xia, Zhefeng Wang, Xinyu Duan, and Baoxing Huai. 2025. A3: Attention-Aware Accurate KV Cache Fusion for Fast Large Language Model Serving. In *arXiv.org*.

[65] Yuxuan Zhu, Ali Falahati, David H. Yang, and Mohammad Mohammadi Amiri. 2025. SentenceKV: Efficient LLM Inference via Sentence-Level Semantic KV Caching. In *arXiv.org*.