IN-MODEL CONTEXT COMPRESSION WITH AGENTIC SOFT TOKENS FOR LONG-HORIZON LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon LLM agents play an important role in solving complex and sustained tasks, with ever-growing interaction contexts. However, these accumulated contexts will raise substantial computational cost and obscure key decision-relevant evidence. Existing methods typically train external compressors or context-management agents to rewrite context as shorter hard-token sequences, but such external selection may not align with the LLM's evolving representation and is necessarily lossy. In this paper, we introduce **Agentic Soft-Token Compression (ASTC)**, a training-free compression method using soft tokens within LLM agent for long-horizon tasks. Specifically, ASTC uses the model's native recurrent dynamics to determine the already absorbed information, leveraging the commonly-used hybrid Gated Delta Network (GDN)–attention. It then carries this information forward as agentic soft tokens while retaining exact values as hard tokens. The resulting compact sequence reduces later-layer computation and yields reusable KV caches, extending the savings across the agent's trajectory. Extensive experiments on long-horizon tasks show that ASTC matches or exceeds the uncompressed agent's accuracy with up to 50.68% fewer end-to-end Attention Pairs, while trained compressors require 15.4×–103.7× more Attention Pairs per successful task. These findings highlight agentic soft tokens as a model-native way to carry absorbed context forward as the task representation evolves. Our code is available at `https://anonymous.4open.science/r/agentic-soft-token-8e3b6a/`.

1 INTRODUCTION

Large language models (LLMs) are increasingly deployed as agents that execute multi-step workflows rather than produce isolated responses. Each step appends system instructions, skills, earlier decisions, tool interactions, code, and environmental feedback to the next model call. The resulting context connects decisions across time, so losing an early constraint or observation can trigger an inconsistent action whose effects propagate through the remaining trajectory. Preserving the full context, however, increases prefill and attention computation, enlarges the key–value (KV) cache, and forces relevant evidence to compete with stale or redundant information. Long-horizon agents therefore face the central challenge of controlling this growing cost without disrupting the cross-step information flow required for coherent decisions.

Reducing context under this constraint is not an ordinary summarization problem. Agent context mixes semantic content with exact values. Explanations, code, and repeated observations can become redundant once the acting LLM has integrated their information, whereas identifiers, API arguments, and numerical values may determine later actions and must remain exact. Information that has little effect on the current action may also become decisive several steps later. Such delayed relevance makes one-shot importance estimates unreliable. Compressibility therefore cannot be inferred from length, repetition, or semantic similarity alone. It depends on what the acting LLM has already absorbed, what the current decision requires, and what a future step may need to recover. Because compression recurs as the trajectory grows, each operation must save more computation than it consumes. Effective context compression must therefore compact information already absorbed by the acting LLM, preserve exact or potentially useful evidence as the task evolves, and deliver savings across subsequent decoding and agent steps.

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

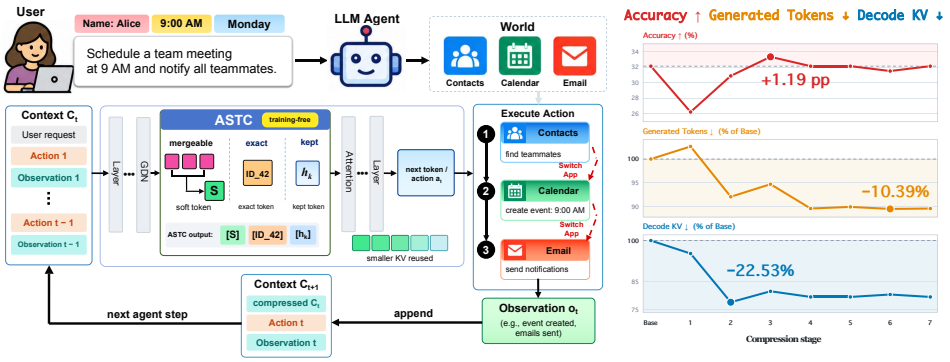


Figure 1: ASTC in a long-horizon agent and in-model stage sensitivity on AppWorld.

Existing context-compression methods typically rely on training, using either hard-token or soft-token compression. Hard-token methods prune or rewrite text, risking irreversible loss of exact values (Jiang et al., 2023; Pan et al., 2024; Xu et al., 2024a; Kang et al., 2026; Liu et al., 2026). Soft-token methods preserve continuous representations but require separately trained encoders or compressors (Mu et al., 2023; Chevalier et al., 2023; Ge et al., 2024; Li et al., 2026). More fundamentally, separately trained compressors create a *model-taste gap*. The information they learn to preserve may not match what the acting LLM has already absorbed or will need later. This motivates an in-model approach that uses the acting LLM's recurrent state to determine compressibility and carries absorbed information forward as agentic soft tokens.

In this paper, we introduce **Agentic Soft-Token Compression (ASTC)**, a training-free approach that uses the acting LLM's own computation to decide what its growing context must retain. The early layers have read the full context and the later layers would otherwise continue to process every original token. Figure 1 places this boundary within a continuing agent workflow and shows how the choice of compression stage changes task accuracy and workload. In hybrid Gated Delta Network (GDN)–attention models (Yang et al., 2025), ASTC examines native recurrent updates and checks whether the retained state can recover a token's contribution. It keeps exact values explicit as hard tokens and carries eligible content forward as agentic soft tokens. The shorter sequence reduces attention work for the current prediction and creates smaller KV caches that remain reusable as the agent continues. Each compression decision is therefore informed by what the model has already absorbed, and its benefit extends across the trajectory.

Across diverse long-horizon tasks, ASTC maintains task quality close to or above the uncompressed model while reducing end-to-end Attention Pairs by up to 50.68%. Preserving exact values avoids the sharp accuracy loss seen when protection is removed, while the compact cache continues to save attention work over later decisions with only small changes in action distributions. The same training-free criterion extends beyond tool observations, matching uncompressed accuracy with 24.52% fewer Decode KV interactions under full-history compression and reducing workload in additional long-context and code-repair tasks. ASTC combines the highest task accuracy with the lowest Attention Pairs per successful task, a 15.4×–103.7× advantage over trained compressors.

Our contributions are summarized as follows:

- We identify the core challenge of long-horizon compression as reducing context cost without losing future-relevant evidence or diverging from the acting LLM's task representation.
- We introduce ASTC, a training-free method that uses native recurrent dynamics to retain exact values as hard tokens and carry absorbed content forward as soft tokens with reusable compact KV caches across subsequent agent steps.
- We bound trained-compressor objective mismatch, establish conditions under which ASTC has lower next-token distortion than equal-length hard-token representations, and quantify attention savings from cache reuse.
- Across long-horizon tasks, ASTC achieves comparable or better accuracy with up to 50.68% fewer end-to-end Attention Pairs and a 15.4×–103.7× advantage in end-to-end Attention Pairs per successful task over trained compressors.

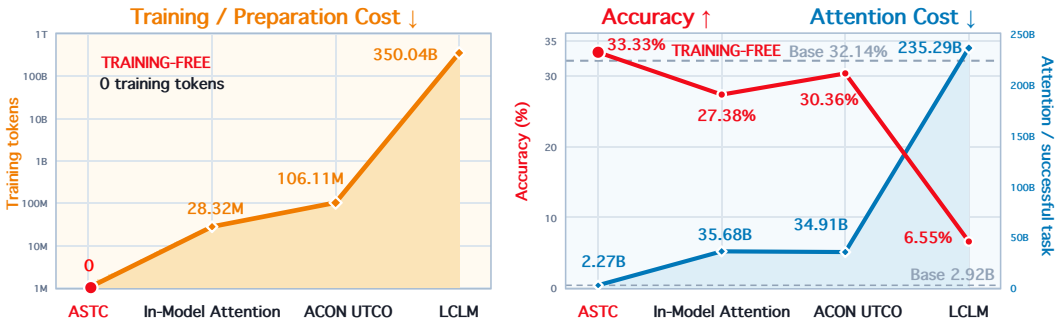


Figure 2: Training/preparation cost and AppWorld test results.

2 Motivation

2.1 In-Model Soft Tokens

Context compression determines both how many tokens remain and which evidence reaches the acting LLM. Existing methods make this decision before the model processes the compressed context, either pruning tokens or rewriting earlier interactions as shorter text (Jiang et al., 2023; Pan et al., 2024; Kang et al., 2026; Liu et al., 2026). This ordering creates a selection mismatch because the value of a context span is not intrinsic to its text. It depends on how the acting LLM represents the current task, what it has already absorbed, and which action it is about to take; the same observation can be redundant at one step and decisive at another. An external compressor must approximate this model-dependent boundary from text alone. In a long-horizon trajectory, a mistaken deletion can propagate because later decisions are made from the retained context.Hard-token output adds a second bottleneck because retained information must be serialized as a shorter sequence of vocabulary tokens, so exact structure or features distributed across positions must be reconstructed as short text. Even a correctly selected span can therefore lose task-relevant information when forced into a short textual rewrite. Moving compression inside the model lets the acting LLM judge compressibility where task information is represented, while soft tokens retain information in continuous contextual representations. In-model selection and soft-token representation are therefore complementary, letting the acting LLM judge what matters and preserve that information in its own representation space, thereby avoiding both external selection mismatch and discrete textual reconstruction.

2.2 Tool Observations

We analyze agent trajectories from real-world user interactions across systems such as Fable 5, GPT-5.6 Sol, Kimi K3, and GLM-5.2 (Havens, 2026a;c;d;b). As Figure 3 shows, a consistent pattern emerges across these trajectories. Tool observations occupy the largest share of the accumulated context and continue to grow as the agent interacts with its environment. In our experiments, tool observations are the primary compression target.

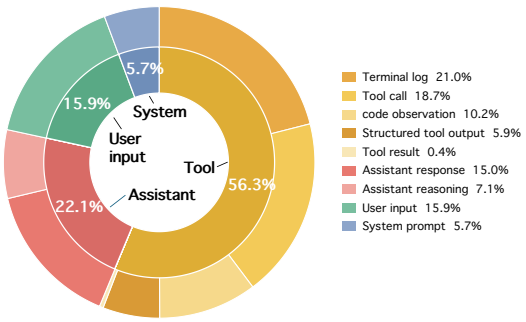


Figure 3: Composition of accumulated context in real-world agent trajectories.

Tool observations are difficult to compress because they mix compressible natural-language and code spans with exact values, such as paths, identifiers, and checksums, that later actions may consume verbatim. Our preliminary experiments show that the former tolerates aggressive compression, whereas changing a single character in the latter can invalidate a later action. This heterogeneity makes tool observations a demanding but informative testbed. Effective agent compression must therefore compact semantic information distributed across many tokens without altering the exact values on which future actions depend.

2.3 Model-Native Selection

Our initial approach followed existing work by training a separate in-model compressor to produce soft tokens. We inserted an attention–MLP compressor after either a single backbone layer or multiple layers, thereby compressing the hidden sequence once or repeatedly, and trained it with **On-policy distillation (OPD)**. Comparing these variants with ACON and LCLM reveals a consistent pattern: training-based compression incurs substantial preparation cost yet delivers weak task performance, while repeated intervention across more layers offers no remedy (Figure 2) (Kang et al., 2026; Li et al., 2026). This failure reflects a difficult optimization problem. An auxiliary compressor must infer what the acting LLM has already absorbed, retain information that later steps may require, and emit features compatible with downstream layers, all through an objective separate from the LLM's native prediction process.

Hybrid GDN–attention models already compute a signal closely aligned with this decision during their normal forward pass. A GDN recurrent state acts as an online associative memory. Its gated delta update can be written as (Yang et al., 2025)

$$\mathbf{S}_t = \alpha_t \mathbf{S}_{t-1} + \beta_t \mathbf{r}_t \mathbf{k}_t^\top, \qquad \mathbf{r}_t = \mathbf{v}_t - \alpha_t \mathbf{S}_{t-1} \mathbf{k}_t, \tag{1}$$

where α_t controls how much previous memory survives, β_t controls the write strength, and $\mathbf{r}_t$ is the part of the incoming value that cannot be recovered from the retained state. Information already represented by the state produces a small correction, whereas novel or conflicting information changes the memory more strongly. The update therefore measures information relative to what the acting LLM currently remembers, not importance inferred from surface form alone. Moreover, the gate and delta update are optimized jointly with the LLM's predictions, so this distinction shares the representation space and task state used by downstream layers. Reusing GDN dynamics avoids teaching an auxiliary compressor to approximate the acting LLM's own notion of novelty. Training-free selection is therefore not merely a cheaper alternative; it follows from recognizing that the acting LLM has already performed the information judgment that compression requires.

3 Related Work

Discrete context compression shortens the sequence before it reaches the acting LLM. Selective Context (Li et al., 2023) prunes redundant spans; LLMLingua (Jiang et al., 2023) and LongLLMLingua (Jiang et al., 2024) rank tokens for prompt pruning; LLMLingua-2 (Pan et al., 2024) learns selection from distilled labels; and RECOMP (Xu et al., 2024a) trains compressors for retrieved evidence. For extended agent workflows, ACON (Kang et al., 2026) optimizes textual compression guidelines from task failures, while Context as a Tool (CAT) (Liu et al., 2026) trains SWE-Compressor to invoke context management during execution. MEM1 (Zhou et al., 2026) instead trains a compact memory jointly with agent reasoning. While these lines reduce accumulated context, text-level selection can discard exact evidence or diverge from the acting LLM's evolving task representation, and learned memory policies require additional training.

Soft-token methods retain information in continuous states. Gist Tokens (Mu et al., 2023) learns compressed prompt activations, AutoCompressors (Chevalier et al., 2023) produce recursive summary vectors, and ICAE (Ge et al., 2024) trains compact memory slots. Sentinel Tokens (Ren et al., 2023) and Activation Beacon (Zhang et al., 2025) compress intermediate activations for later processing. DAST (Chen et al., 2025) allocates soft tokens according to contextual information density, while Latent Context Language Models (LCLMs) (Li et al., 2026) train dedicated encoder–decoder models for large-scale compression. For tool-using LMs, Xu et al. (2024b) combine compressed representations with raw tokens for exact tool interfaces. These studies show that continuous memory can support incremental reuse and coexist with exact-token retention. Their compression behavior is nevertheless learned through model adaptation or auxiliary components, and their primary settings differ from a growing history of agent actions and observations.

ASTC brings the continuous memory studied in soft-token compression to the growing histories addressed by agent context-management work. By operating within the acting LLM without a separately trained compressor, it keeps compression tied to the model's evolving task representation. To the best of our knowledge, ASTC is among the first in-model soft-token methods designed for context compression in long-horizon LLM agents.

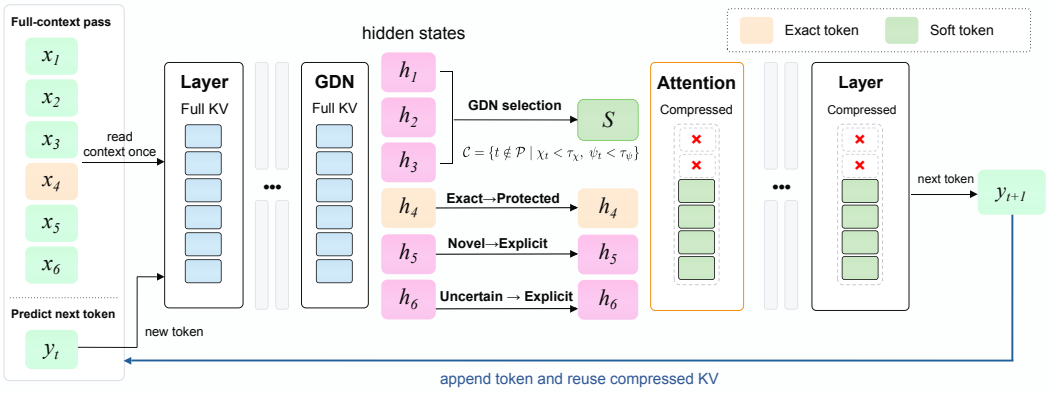


Figure 4: Overview of ASTC for in-model context compression and KV cache reuse.

4 METHOD

4.1 AGENTIC SOFT-TOKEN COMPRESSION

At step a, early layers process every token in the uncached suffix $\Delta X^{(a)}$, using cache $\mathcal{K}_{<}^{(a-1)}$ for prior context. ASTC then compresses the new hidden states in tool-observation spans $\mathcal{O}$ subject to protection mask P:

$$\begin{aligned} H_{\ell^\star}^{\text{new}} &= F_{0:\ell^\star}(\Delta X^{(a)}; \mathcal{K}_{<}^{(a-1)}) \in \mathbb{R}^{m \times d}, \\ \widetilde{H}_{\ell^\star}^{\text{new}} &= \mathcal{C}(H_{\ell^\star}^{\text{new}}; \mathcal{O}, P) \in \mathbb{R}^{\widetilde{m} \times d}, \quad \widetilde{m} \le m. \end{aligned} \tag{2}$$

P keeps recognized exact values and the action boundary as hard tokens, although it may miss some exact fields. For each GDN head h, let $\mathbf{S}_{i-1,h}$ be its state before position i, with key $\mathbf{k}_{i,h}$, value $\mathbf{v}_{i,h}$, and decay $\alpha_{i,h} = \exp(g_{i,h})$. Let $\mathbf{v}_i$ and $\boldsymbol{\delta}_i$ concatenate the head-wise vectors. The first test measures innovation relative to the recurrent state:

$$\boldsymbol{\delta}_{i,h} = \mathbf{v}_{i,h} - \alpha_{i,h}\, \mathcal{R}(\mathbf{S}_{i-1,h}, \mathbf{k}_{i,h}), \qquad \chi_i = \frac{\mathrm{RMS}(\boldsymbol{\delta}_i)}{\mathrm{RMS}(\mathbf{v}_i) + \epsilon}, \tag{3}$$

where $\mathcal{R}(\mathbf{S}, \mathbf{k}) = \mathbf{S}\mathbf{k}$. A small χ_i means the incoming value was largely predicted by the preceding state. Prediction at position i does not ensure that later updates retain the value, so ASTC also reads the observation-end state $\mathbf{S}_{o,h}$ with a fixed, untrained spectral basis $\mathbf{Q}_h \in \mathbb{R}^{d_{k,h} \times M}$, using the pooling slot budget M:

$$\widehat{\mathbf{v}}_{i,h} = \mathbf{S}_{o,h}\mathbf{Q}_h(\mathbf{Q}_h^\top \mathbf{Q}_h + \lambda \mathbf{I})^{-1}\mathbf{Q}_h^\top \mathbf{k}_{i,h}, \qquad \psi_i = \frac{\mathrm{RMS}(\mathbf{v}_i - \widehat{\mathbf{v}}_i)}{\mathrm{RMS}(\mathbf{v}_i) + \epsilon}. \tag{4}$$

Here $\widehat{\mathbf{v}}_i$ concatenates the per-head reads. ASTC selects an unprotected position only when it is both locally predictable ($\chi_i < \tau_\chi$) and still readable at the endpoint ($\psi_i < \tau_\psi$).

The selected positions still need a compact representation. ASTC partitions them into at most M causal groups, trims outliers around a member of each group, and mean-pools the retained states of group G at the boundary:

$$\mathbf{z}_G = \mathrm{cast}\left(\frac{1}{|G|} \sum_{i \in G} \mathbf{h}_i^{\mathrm{FP32}} \right). \tag{5}$$

The replacement stays in the acting model's hidden space rather than being rewritten as text. Only multi-position groups with maximum post-cast relative hidden-state error at most τ_h become *agentic soft tokens*. Protected, rejected, and unmerged positions remain hard tokens. Each soft token takes its group's latest logical position, and a stable sort restores causal order.

Compression leaves the pre-boundary caches at their original-token lengths, while downstream attention caches the shorter mixed sequence. This cache is reused during decoding and later agent steps, so an accepted merge saves work beyond the prefill in which it occurs. Algorithm 1 specifies the per-observation commit gate; bounded pooling is detailed in Appendix C.1.

Algorithm 1 ASTC at one prefill step with a persistent hybrid cache

Require: Uncached suffix $\Delta X^{(a)}$, existing layer-specific hybrid cache $\mathcal{K}^{(a-1)}$, closed observation spans $\mathcal{O}$ in this suffix, protection mask P, frozen F, boundary $\ell^\star$, reader $\mathbf{Q}$, thresholds $\tau_\chi, \tau_\psi, \tau_h$, group cap M, budget B, remaining-layer count L_{rem}, fraction ρ

Ensure: Next-token logits; updated hybrid cache $\mathcal{K}^{(a)}$

for each observation $o \in \mathcal{O}$ **do**
  $\mathcal{E}_o \leftarrow \{i \in o : P_i = 0 \text{ and } i \text{ is an original position}\}$
  $r_o \leftarrow \max(\lceil B/L_{\mathrm{rem}} \rceil, \lceil \rho |\mathcal{E}_o| \rceil)$
  Bypass o if it already contains a soft token or $|\mathcal{E}_o| < r_o + 1$
end for
Run $F_{0:\ell^\star}$ on $\Delta X^{(a)}$ with its existing cache; update early-layer caches and obtain $H^{\mathrm{new}}_{\ell^\star}$
During this run, capture native GDN tensors and read endpoint states only for non-bypassed spans
for each non-bypassed observation $o \in \mathcal{O}$ **do**
  $\mathcal{C}_o \leftarrow \{i \in \mathcal{E}_o : \chi_i < \tau_\chi\}$
  Continue to the next observation if $|\mathcal{C}_o| < r_o + 1$
  $\mathcal{C}_o \leftarrow \{i \in \mathcal{C}_o : \psi_i < \tau_\psi\}$
  Continue to the next observation if $|\mathcal{C}_o| < r_o + 1$
  $(\mathcal{M}_o, \mathcal{Z}_o) \leftarrow \textsc{BoundedPool}(H^{\mathrm{new}}_{\ell^\star}[\mathcal{C}_o], M, \tau_h)$
  if $|\mathcal{M}_o| - |\mathcal{Z}_o| \geq r_o$ **then**
    Replace $\mathcal{M}_o$ by $\mathcal{Z}_o$ at each group's greatest logical position
  end if
end for
Stably sort the mixed new suffix by logical position
Run $F_{\ell^\star+1:L}$ on that suffix with its existing cache; update downstream attention KV and GDN recurrent and convolutional states
Apply final norm and LM head to the last protected original position
return next-token logits and $\mathcal{K}^{(a)}$

Here B is the minimum downstream token–layer reduction, L_{rem} counts layers after the boundary, and ρ is the minimum reduction as a fraction of eligible positions in each observation.

4.2 Theoretical Analysis

We examine what native selection can certify, how a compact cache saves work over time, and how merging can shift downstream attention. The distributional claims below concern a fixed next-token prefix; they do not imply identical agent trajectories.

4.2.1 Model-Native Selection and Continuous Representation

Fix a history, a next-token prefix, and a boundary $\ell^\star$. Let H be the full boundary sequence and Z a length-k replacement at fixed logical positions. We measure the acting model's next-token distortion by

$$D_\theta(H, Z) = D_{\mathrm{KL}}\Big(p_\theta^{>\ell^\star}(\cdot \mid H) \,\|\, p_\theta^{>\ell^\star}(\cdot \mid Z)\Big). \tag{6}$$

A separately trained compressor minimizes a surrogate objective J_{tr}, which need not match D_θ. On the same nonempty admissible set $\mathcal{A}_k$, if both minima are attained, $\sup_{Z \in \mathcal{A}_k} |D_\theta(H, Z) - J_{\mathrm{tr}}(H, Z)| \leq \varepsilon$, and the returned Z_{tr} is within ξ of the surrogate minimum, then

$$0 \leq D_\theta(H, Z_{\mathrm{tr}}) - \min_{Z \in \mathcal{A}_k} D_\theta(H, Z) \leq 2\varepsilon + \xi. \tag{7}$$

This decomposition exposes a possible source of error, but does not establish a nonzero gap or rank methods with different feasible sets.

ASTC instead reads the acting model's own GDN signals. The update in Equation 1 yields a per-head write bound proportional to χ_i, while ψ_i measures what remains readable at the observation endpoint. These tests describe state coverage, but replacing positions can also shift downstream attention and recurrent states. For a fixed next-token prefix, suppose those effects admit aligned local perturbations e_j and the remaining logit map is L_j-Lipschitz at each site. Write $\mathbf{l}$ and $\widetilde{\mathbf{l}}$ for their logits, and p and $\widetilde{p}$ for their next-token distributions. With $E = \sum_j L_j e_j$, we obtain

$$\|\widetilde{\mathbf{l}} - \mathbf{l}\|_\infty \leq E, \qquad D_{\mathrm{KL}}(p \| \widetilde{p}) \leq 2E. \tag{8}$$

Hence the next-token distribution changes little when E is small, and the greedy token is unchanged if its original logit margin exceeds $2E$. The selector tests alone do not bound every term in E.

Stability addresses whether a replacement is safe; the representation space determines what it can retain at a fixed budget. At equal physical length k and fixed logical positions in an unchanged context, hard-token reachable boundary states form a finite set $\mathcal{A}_k^{\mathrm{hard}}$ inside the continuous space $\mathcal{A}_k^{\mathrm{soft}} = \mathbb{R}^{k\times d}$. With full-context distribution $p_H = p_\theta^{>\ell^\star}(\cdot \mid H)$, let $\delta_k(H) = \min_{Z\in\mathcal{A}_k^{\mathrm{hard}}} \|p_H - p_\theta^{>\ell^\star}(\cdot \mid Z)\|_1$. Set inclusion and Pinsker's inequality give

$$\begin{aligned} \inf_{Z\in\mathcal{A}_k^{\mathrm{soft}}} D_\theta(H,Z) &\le \min_{Z\in\mathcal{A}_k^{\mathrm{hard}}} D_\theta(H,Z), \\ D_\theta(H,Z_{\mathrm{ASTC}}) &< \min_{Z\in\mathcal{A}_k^{\mathrm{hard}}} D_\theta(H,Z) \quad \text{if } E < \delta_k(H)^2/4. \end{aligned} \tag{9}$$

The first inequality describes representational capacity. The second gives a sufficient condition for ASTC's particular soft-token output to outperform every length-k hard-token representation. Its premise is not guaranteed by the selector thresholds; proofs appear in Appendix A.

4.2.2 One-Pass and Amortized Efficiency

The boundary also determines when savings arise. Early layers still read the new block, but later attention sees fewer positions in the same pass; the avoided work must exceed selection and pooling cost for a net gain. Once the shorter KV cache persists, subsequent queries revisit fewer keys.

Theorem 4.1 (Amortized attention-pair reduction)**.** *Suppose the affected downstream layers contain $A_>$ full causal attention layers. Before one commit, the current prefill has physical length n_{phys}, including an unchanged cached prefix, and all r removed positions belong to the newly processed block. Under the same continuation of N_q later queries reaching these layers, assume the compared caches retain their r-position difference. Then the total reduction in query–key token pairs is*

$$\Delta Q = A_> r \left(\frac{2n_{\mathrm{phys}} - r + 1}{2} + N_q \right). \tag{10}$$

The first term counts the current prefill, while N_q counts reuse of the same compact cache. Savings therefore accumulate without another compression decision. The proof in Appendix A.6 concerns attention pairs rather than wall-clock speedup.

4.2.3 Downstream Attention Reweighting

Compression can also change which evidence downstream attention emphasizes. A redundant group may attract attention through multiplicity alone. The following idealized lemma isolates that effect.

Lemma 4.2 (Attention mass under idealized merging)**.** *For the same downstream query in both runs, after the last logical position of a merged group G ($|G| = g > 1$), assume its members and representative share score s and value $\mathbf{v}_G$. Let retained positions have unchanged scores and values, with unnormalized mass $Z_R > 0$ and attention-weighted value $\overline{\mathbf{v}}_R$. Their total attention masses before and after replacement, A_R and $\widetilde{A}_R$, satisfy*

$$\widetilde{A}_R - A_R = \frac{Z_R(g-1)e^s}{(Z_R + e^s)(Z_R + ge^s)} > 0, \qquad \mathbf{u}^\top(\widetilde{\mathbf{o}} - \mathbf{o}) = (\widetilde{A}_R - A_R)\mathbf{u}^\top(\overline{\mathbf{v}}_R - \mathbf{v}_G) \tag{11}$$

for any linear readout direction $\mathbf{u}$.

The group retains one representative but loses the extra attention weight of its $g-1$ copies. If the retained value has a larger projection along $\mathbf{u}$ than $\mathbf{v}_G$, the attention output shifts toward it. This describes an idealized local effect, not a guarantee of improved task accuracy. Appendix A.7 proves the lemma and bounds the unequal-score and unequal-value cases separately.

5 Experiments

We evaluate ASTC with a frozen Qwen3.5-9B acting model on AppWorld (Trivedi et al., 2024), OfficeBench (Wang et al., 2024), and 8-Objective QA (Zhou et al., 2026; Kang et al., 2026).

We compare with uncompressed Base, prompted compression, LLMLingua (Jiang et al., 2023), LLMLingua-2 (Pan et al., 2024), ACON UT and UTCO (Kang et al., 2026), and LCLM (Li et al., 2026). Here, ASTC compresses only eligible tool observations, not the full context. We measure task quality and end-to-end Decode KV, Attention Pairs, and Generated Tokens. Decode KV counts decoding query–cache interactions; Attention Pairs counts causal query–key pairs across prefill, decoding, and auxiliary calls. Detailed setups and additional experiments are in Appendices C–G.

5.1 Main Results

Table 1: Accuracy and end-to-end inference workload on three long-horizon agentic benchmarks.

		AppWorld				**OfficeBench**				**8-Objective QA**				
Method	Training-free	Acc.% ↑	D-KV% ↓	Attn.% ↓	Gen.% ↓	Acc.% ↑	D-KV% ↓	Attn.% ↓	Gen.% ↓	EM% ↑	F1% ↑	D-KV% ↓	Attn.% ↓	Gen.% ↓
Base	–	<u>32.14</u>	100.00	100.00	100.00	48.42	100.00	100.00	100.00	28.28	37.34	100.00	100.00	100.00
Hard-token compression														
Prompting	✓	25.00	143.59	**64.79**	104.25	**52.63**	83.09	<u>77.32</u>	91.42	26.89	36.61	97.36	26.68	232.78
LLMLingua v1	✗	5.36	131.00	159.75	99.81	24.21	259.99	319.34	205.51	14.39	20.55	104.24	43.38	110.62
LLMLingua-2	✗	7.14	225.53	3,364.23	<u>83.22</u>	44.21	69.59	961.40	100.50	24.12	33.56	49.69	25.37	87.52
ACON UT	✗	29.17	261.91	1,064.69	257.49	<u>49.47</u>	163.11	1,180.93	135.36	27.53	37.34	68.82	16.01	187.74
ACON UTCO	✗	30.36	243.82	1,130.67	255.95	48.42	80.77	1,197.57	90.43	<u>28.54</u>	<u>37.40</u>	<u>42.99</u>	<u>14.62</u>	124.98
Soft-token compression														
LCLM	✗	6.55	**23.94**	1,643.65	**35.08**	21.05	**20.83**	641.30	**33.37**	0.00	0.00	**22.47**	**0.83**	**0.20**
ASTC	✓	**33.33**	<u>81.55</u>	<u>80.70</u>	94.83	47.37	<u>54.52</u>	**49.32**	<u>69.23</u>	**30.56**	**38.46**	78.69	98.51	<u>86.55</u>

Note. Workload includes auxiliary compressor calls, including LLMLingua, and is relative to Base (100%). **Bold** and <u>underlined</u> mark the best and second-best values by arrow.

Table 1 shows that ASTC offers a favorable balance of task quality and inference workload under the distinct interaction demands of the three benchmarks. On AppWorld, it is the only evaluated method that improves accuracy over Base while reducing all three workload measures; competing compressors obtain isolated savings but sacrifice accuracy or increase work elsewhere. On OfficeBench, ASTC remains close to Base in accuracy while substantially reducing Decode KV and Attention Pairs; Prompting achieves higher accuracy but uses more Decode KV and generated tokens than ASTC. On 8-Objective QA, ASTC achieves the best answer quality while reducing generated tokens. LCLM has the lowest measured workload but answers none of the tasks correctly; ACON UTCO reduces Decode KV and Attention Pairs further than ASTC but has lower answer quality. Taken together, these results place ASTC's value at the trajectory level, where retained task quality and cumulative inference workload must be judged together.

5.2 Model-Native Selector Controls

We ask whether ASTC's GDN signals improve selection beyond what protection and sequence shortening provide. We test this by retaining the acting model, protection rules, and legal group construction while changing the selection rule. Hidden-only uses hidden-state reconstruction error to test whether local fidelity suffices; Random assigns fixed random scores to test whether protected grouping alone explains the gains. Each policy runs the full AppWorld in closed loop, allowing compression choices to shape later actions, tool observations, and workload. Figure 5 compares task success with end-to-end Attention Pairs and shows ASTC's strongest joint outcome, consistent with the value of model-native selection. Because trajectories and compression events differ, the comparison evaluates complete policies; paired bootstrap uncertainty makes the success advantage suggestive rather than conclusive. The comparison suggests that the acting model's accumulated state supplies information about compressibility that local reconstruction alone does not capture.

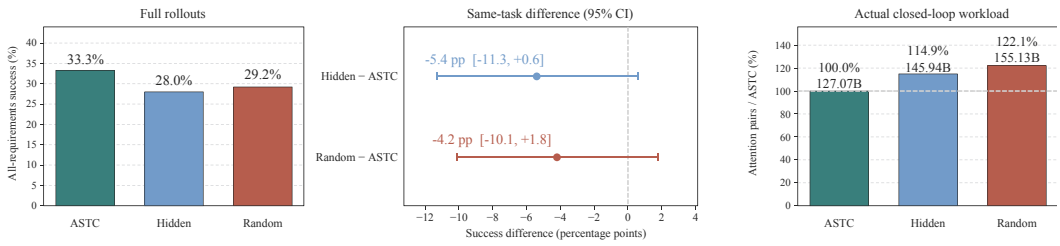


Figure 5: Selector controls on AppWorld.

5.3 Persistence After One Commit

Compression helps a long-horizon agent only if its shorter KV cache remains useful when later steps revisit it, without substantially changing the agent's decisions. To isolate this mechanism, we branch from archived Base trajectories, allow at most one ASTC commit in one branch, and replay identical subsequent action tokens and tool results in both. This fixed continuation separates the effect of persistent cache compression from changes caused by diverging actions or observations. With no further commits allowed, savings at later horizons reflect reuse of the initial cache update rather than repeated compression. At successive action horizons during the shared continuation, we measure KL divergence between conditional action-token distributions and the Attention Pairs saved by reusing the shorter cache. Figure 6 shows that the distributions remain close while savings recur at later actions, supporting the proposed amortization across the trajectory. A separate autonomous one-commit run probes whether this local stability carries through to task outcomes; its success is somewhat lower than archived Base and full ASTC, and differing archive versions make the comparison descriptive. Under a fixed continuation, the value of one memory update accumulates as later actions repeatedly reuse the compact cache it created.



Figure 6: One-commit cache reuse and task outcomes.

5.4 Component Ablations

To identify which parts of ASTC sustain its accuracy–efficiency balance, we remove protection, either GDN selection test (χ or ψ), or soft-token replacement one at a time on the same AppWorld task set. Table 2 shows that removing protection sharply lowers accuracy despite reducing Attention Pairs, indicating why exact values must remain explicit. Removing either selection test or using hard deletion also lowers accuracy and raises Decode KV and Generated Tokens relative to full ASTC. These ablations show how model-native selection, exact-value protection, and soft-token retention work together to preserve useful context while reducing inference work.

Table 2: Component ablations on AppWorld.

Variant	Acc. ↑	D-KV ↓	Attn. ↓	Gen. ↓
Base	32.14%	100.00%	100.00%	100.00%
ASTC	**33.33%**	**81.55%**	80.70%	**94.83%**
No protection	20.83%	98.46%	**68.63%**	127.05%
No χ	32.14%	84.79%	78.11%	97.76%
No ψ	32.74%	89.72%	86.08%	99.65%
Hard deletion	32.74%	88.83%	86.85%	99.58%

6 Conclusion

Long-horizon agents must retain evidence for future decisions as context grows, while separately trained compressors may drift from the acting LLM's evolving task representation. ASTC uses native recurrent dynamics after early layers have read the full sequence, carrying absorbed content as agentic soft tokens and keeping exact values as hard tokens. The shorter sequence cuts later-layer computation and creates a compact KV cache whose savings persist across subsequent decisions. Without compressor training, ASTC keeps accuracy close to or above the uncompressed agent with up to 50.68% fewer end-to-end Attention Pairs; in the controlled comparison, it attains the highest accuracy and lowest Attention Pairs per successful task. Ablations and fixed-continuation analysis link these gains to exact-value protection, model-native selection, and persistent cache reuse. These results show how the acting LLM can compact absorbed context into reusable state without losing exact evidence for later decisions.

AI USE STATEMENT

Generative AI tools were used to polish the language and presentation of author-provided manuscript text, including rephrasing and reorganizing sections for clarity. The authors reviewed all AI-assisted edits and take full responsibility for the paper's content.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. URL https://arxiv.org/abs/2108.07732.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*, 2024. URL https://arxiv.org/abs/2412.15204.

Shaoshen Chen, Yangning Li, Zishan Xu, Yongqin Zeng, Shunlong Wu, Xinshuo Hu, Zifei Shan, Xin Su, Jiwei Tang, Yinghui Li, and Hai-Tao Zheng. DAST: Context-aware compression in LLMs via dynamic allocation of soft tokens. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20544–20552, 2025. doi: 10.18653/v1/2025.findings-acl.1055. URL https://aclanthology.org/2025.findings-acl.1055/.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3829–3846, 2023. doi: 10.18653/v1/2023.emnlp-main.232. URL https://aclanthology.org/2023.emnlp-main.232.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uREj4ZuGJE.

Greg Havens. Claude Fable 5 Agent Traces. Hugging Face dataset, 2026a. URL https://huggingface.co/datasets/greghavens/fable-5-coding-and-debugging-traces-synthetic-corrections.

Greg Havens. GLM 5.2 Agent Traces. Hugging Face dataset, 2026b. URL https://huggingface.co/datasets/greghavens/glm-5.2-coding-and-debugging-traces.

Greg Havens. GPT-5.6 Sol coding and debugging traces. Hugging Face dataset, 2026c. URL https://huggingface.co/datasets/greghavens/gpt-5.6-sol-coding-and-debugging-traces.

Greg Havens. Kimi K3 coding, tool use, and instruction following traces. Hugging Face dataset, 2026d. URL https://huggingface.co/datasets/greghavens/kimi-k3-coding-and-debugging-traces.

Xuning He, Zinan Sheng, Yongding Tao, Huanyu Liu, Ge Li, Xue Jiang, and Yihong Dong. Archer: Adaptive reuse of cached hidden states for efficient rollback in diffusion language models. *arXiv preprint arXiv:2608.08086*, 2026. URL https://arxiv.org/abs/2608.08086.

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

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376, 2023. doi: 10.18653/v1/2023.emnlp-main.825. URL https://aclanthology.org/2023.emnlp-main.825.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677, 2024. doi: 10.18653/v1/2024.acl-long.91. URL https://aclanthology.org/2024.acl-long.91.

Minki Kang, Wei-Ning Chen, Dongge Han, Huseyin A. Inan, Lukas Wutschitz, Yanzhi Chen, Robert Sim, and Saravan Rajmohan. ACON: Optimizing context compression for long-horizon LLM agents. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.00615.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Ang Li, Sean McLeish, Haozhe Chen, Nimit Kalra, Zaiqian Chen, Artem Gazizov, Venkata Anoop Suhas Kumar Morisetty, Bhavya Kailkhura, Harshitha Menon, Zhuang Liu, Brian R. Bartoldson, Tom Goldstein, Sanae Lotfi, Micah Goldblum, and Pavel Izmailov. End-to-end context compression at scale. *arXiv preprint arXiv:2606.09659*, 2026. URL https://arxiv.org/abs/2606.09659.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6342–6353, 2023. doi: 10.18653/v1/2023.emnlp-main.391. URL https://aclanthology.org/2023.emnlp-main.391/.

Shukai Liu, Bo Jiang, Jian Yang, Yizhi Li, Jinyang Guo, Xianglong Liu, and Bryan Dai. Context as a tool: Context management for long-horizon SWE-agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 20604–20617, 2026. doi: 10.18653/v1/2026.findings-acl.1032. URL https://aclanthology.org/2026.findings-acl.1032/.

Moonshot AI. Kimi-Linear-48B-A3B-Instruct. Hugging Face model card, 2025. URL https://huggingface.co/moonshotai/Kimi-Linear-48B-A3B-Instruct/tree/e1df551a447157d4658b573f9a695d57658590e9.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/3d77c6dcc7f143aa2154e7f4d5e22d68-Abstract.html.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 963–981, 2024. doi: 10.18653/v1/2024.findings-acl.57. URL https://aclanthology.org/2024.findings-acl.57.

Qwen Team. Qwen3.5-9B. Hugging Face model card, 2026. URL https://huggingface.co/Qwen/Qwen3.5-9B/tree/c202236235762e1c871ad0ccb60c8ee5ba337b9a.

Siyu Ren, Qi Jia, and Kenny Zhu. Context compression for auto-regressive transformers with sentinel tokens. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12860–12867, 2023. doi: 10.18653/v1/2023.emnlp-main.794. URL https://aclanthology.org/2023.emnlp-main.794/.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, 2024. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

Zilong Wang, Yuedong Cui, Li Zhong, Zimin Zhang, Da Yin, Bill Yuchen Lin, and Jingbo Shang. OfficeBench: Benchmarking language agents across multiple applications for office automation. *arXiv preprint arXiv:2407.19056*, 2024. URL https://arxiv.org/abs/2407.19056.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *International Conference on Learning Representations*, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/bda88ed2892f5e61c9a9bf215c566913-Abstract-Conference.html.

Yang Xu, Yunlong Feng, Honglin Mu, Yutai Hou, Yitong Li, Xinghao Wang, Wanjun Zhong, Zhongyang Li, Dandan Tu, Qingfu Zhu, Min Zhang, and Wanxiang Che. Concise and precise context compression for tool-using language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 16430–16441, 2024b. doi: 10.18653/v1/2024.findings-acl.974. URL https://aclanthology.org/2024.findings-acl.974/.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations*, 2025. URL https://

arxiv.org/abs/2412.06464.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long context compression with activation beacon. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/fc797c61eb18f7b3ce9a74af0ef0d876-Abstract-Conference.html.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Bryan Kian Hsiang Low, and Paul Pu Liang. MEM1: Learning to synergize memory and reasoning for efficient long-horizon agents. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=XY8AaxDSLb.

A Theoretical Derivations

The distributional comparisons below fix an agent history, a next-token prefix, and a compression boundary while preserving original logical positions. They bound one conditional next-token distribution and do not claim that independently generated trajectories remain identical. The workload proof separately counts later queries while the compressed cache persists, and the attention-mass proof fixes one downstream query.

A.1 Surrogate-Objective Mismatch

Let $\mathcal{A}_k$ be the same nonempty feasible set for both objectives, and assume the minima below exist. Write $D(Z) = D_\theta(H, Z)$ and $J(Z) = J_{\mathrm{tr}}(H, Z)$. If $\sup_{Z\in\mathcal{A}_k} |D(Z) - J(Z)| \le \varepsilon$ and $Z_{\mathrm{tr}} \in \mathcal{A}_k$ satisfies $J(Z_{\mathrm{tr}}) \le \inf_{Z\in\mathcal{A}_k} J(Z) + \xi$, then, for $Z_\theta^\star \in \arg\min_{Z\in\mathcal{A}_k} D(Z)$,

$$\begin{aligned} D(Z_{\mathrm{tr}}) &\le J(Z_{\mathrm{tr}}) + \varepsilon \\ &\le J(Z_\theta^\star) + \xi + \varepsilon \\ &\le D(Z_\theta^\star) + 2\varepsilon + \xi. \end{aligned} \tag{12}$$

Optimality of $Z_\theta^\star$ gives the nonnegative lower endpoint. If $J = D$ and $\xi = 0$, a trained method can attain the optimum. Thus the upper bound separates objective mismatch from optimization error but gives no lower bound on the realized gap and does not order ASTC and trained methods. It also does not apply across unequal feasible sets, such as unrestricted soft states and text-reachable hard states.

A.2 Model-Native Selection Bounds

Write $D_v = \sum_h d_{v,h}$ for the total value width across GDN heads. For each head h, the update $\mathbf{S}_{i,h} = \alpha_{i,h}\mathbf{S}_{i-1,h} + \beta_{i,h}\boldsymbol{\delta}_{i,h}\mathbf{k}_{i,h}^\top$ with $\beta_{i,h} \ge 0$ gives the rank-one identity

$$\|\mathbf{S}_{i,h} - \alpha_{i,h}\mathbf{S}_{i-1,h}\|_F = \beta_{i,h}\|\boldsymbol{\delta}_{i,h}\|_2\|\mathbf{k}_{i,h}\|_2. \tag{13}$$

The implemented RMS aggregates values across heads. Hence $\|\boldsymbol{\delta}_{i,h}\|_2 \le \|\boldsymbol{\delta}_i\|_2 = \sqrt{D_v}\,\chi_i(\mathrm{RMS}(\mathbf{v}_i) + \epsilon)$. Accepting a position with $\chi_i < \tau_\chi$ and $\psi_i < \tau_\psi$ therefore implies

$$\begin{aligned} \|\mathbf{S}_{i,h} - \alpha_{i,h}\mathbf{S}_{i-1,h}\|_F &< \beta_{i,h}\tau_\chi\sqrt{D_v}(\mathrm{RMS}(\mathbf{v}_i) + \epsilon)\|\mathbf{k}_{i,h}\|_2, \\ \|\mathbf{v}_i - \widehat{\mathbf{v}}_i\|_2 &< \tau_\psi\sqrt{D_v}(\mathrm{RMS}(\mathbf{v}_i) + \epsilon). \end{aligned} \tag{14}$$

Both are local to the GDN state and the specified endpoint reader. They do not constrain the attention scores of the same token. Pre-boundary layers process every original token and preserve their recurrent state, so, conditional on the same future input tokens, that state is unchanged by the boundary replacement. The remaining possible perturbation enters through the compressed downstream sequence and its attention caches.

A.3 Downstream Attention Stability

Consider one attention head, the same fixed query vector in both runs after the greatest logical position of a candidate group G, and its replacement. Assume at least one retained position exists and all retained keys and values are unchanged in this local comparison. Let R be the other positions visible to this query, and define $W_G = \sum_{i\in G} e^{s_i}$, $W_R = \sum_{i\in R} e^{s_i}$, and $w_z = e^{s_z}$, where s_i and s_z are the original and replacement attention scores for this query. Define

$$\begin{aligned} a_G &= \frac{W_G}{W_R + W_G}, & b_G &= \frac{w_z}{W_R + w_z}, \\ \bar{\mathbf{v}}_G &= \frac{\sum_{i\in G} e^{s_i}\mathbf{v}_i}{W_G}, & \bar{\mathbf{v}}_R &= \frac{\sum_{i\in R} e^{s_i}\mathbf{v}_i}{W_R}. \end{aligned} \tag{15}$$

The two attention outputs and their difference are exactly

$$\begin{aligned} \mathbf{o} &= (1 - a_G)\bar{\mathbf{v}}_R + a_G\bar{\mathbf{v}}_G, \\ \widetilde{\mathbf{o}} &= (1 - b_G)\bar{\mathbf{v}}_R + b_G\mathbf{v}_z, \\ \widetilde{\mathbf{o}} - \mathbf{o} &= (b_G - a_G)(\mathbf{v}_z - \bar{\mathbf{v}}_R) + a_G(\mathbf{v}_z - \bar{\mathbf{v}}_G). \end{aligned} \tag{16}$$

The triangle inequality gives $\|\widetilde{\mathbf{o}}-\mathbf{o}\|_2 \le |b_G-a_G|\|\mathbf{v}_z-\bar{\mathbf{v}}_R\|_2+a_G\|\mathbf{v}_z-\bar{\mathbf{v}}_G\|_2$. If the value map is L_V-Lipschitz on the group and its representative, and every member is within hidden-state radius r_G of the representative, the second term is at most $a_G L_V r_G$; the attention-mass term remains separate.

The attention-mass condition cannot be omitted. As a counterexample, take $g > 1$ identical group values $\mathbf{v}_G$, one retained value $\mathbf{v}_R \neq \mathbf{v}_G$, and equal attention scores. If the soft representative has value $\mathbf{v}_G$ and the same score, then $a_G = g/(g+1)$ but $b_G = 1/2$, so the two outputs differ despite zero within-group reconstruction error. The GDN innovation and endpoint read error can also be zero in a hybrid model whose attention projections are independent of that state. Thus the native tests alone cannot certify unchanged downstream decisions.

A.4 Downstream Recurrent Stability

Post-boundary GDN layers also consume the shortened sequence. Consider one such layer and align the original and compressed runs at the logical endpoint t_G of a group G. Starting from the same state, let their states there differ by e_G. For later tokens that supply identical keys, values, and gates to this layer, the GDN recurrence has the affine form

$$\mathbf{S}_t = \alpha_t \mathbf{S}_{t-1}(\mathbf{I} - \beta_t \mathbf{k}_t \mathbf{k}_t^\top) + \beta_t \mathbf{v}_t \mathbf{k}_t^\top. \tag{17}$$

Its state map is ρ_t-Lipschitz in Frobenius norm with $\rho_t = \alpha_t \|\mathbf{I} - \beta_t \mathbf{k}_t \mathbf{k}_t^\top\|_2$. Induction therefore gives, after m such tokens,

$$\|\mathbf{S}_{t_G+m} - \widetilde{\mathbf{S}}_{t_G+m}\|_F \le e_G \prod_{u=1}^{m} \rho_{t_G+u}. \tag{18}$$

If $\|\mathbf{k}_t\|_2 \le 1$, $0 \le \beta_t \le 1$, and $0 \le \alpha_t \le 1$, then $\rho_t \le \alpha_t \le 1$. This result is conditional on identical later inputs to that GDN layer. It does not bound the initial defect e_G, which concerns a different downstream layer from the GDN used to compute χ and ψ.

For a full-model statement, assume an alignment between the two finite causal computations at every group endpoint. Assume further that they can be joined by a sequence of hybrids in which each adjacent pair differs at one affected attention output or downstream recurrent state, with all its upstream inputs equal. Let e_j bound that local difference (using the attention bound above or a state-defect bound), and let the map from that quantity to the next-token logits be L_j-Lipschitz on the segment between the two values. Applying the triangle inequality to the successive hybrid logits proves

$$\|\widetilde{\mathbf{l}} - \mathbf{l}\|_\infty \le E := \sum_j L_j e_j. \tag{19}$$

Because log-sum-exp is 1-Lipschitz in the infinity norm, each log-probability changes by at most $2E$ and $D_{\mathrm{KL}}(p\|\widetilde{p}) \le 2E$. If the original largest logit exceeds the second largest by more than $2E$, its index remains the greedy next token. This theorem is conditional: χ, ψ, and the pooling threshold alone do not supply all e_j or L_j. It concerns the fixed next-token prefix and does not extend to independently generated trajectories without further stability assumptions.

A.5 Continuous Representation at a Fixed Budget

Fix k and logical position IDs π at the compression boundary; hold the cached prefix and all untouched boundary states fixed. The notation $F_{0:\ell^\star}(t_{1:k};\pi)$ evaluates all candidate sequences at those same positions. For a finite vocabulary $\mathcal{V}$, the hard-token reachable set $\mathcal{A}_k^{\mathrm{hard}} = \{F_{0:\ell^\star}(t_{1:k};\pi) : t_{1:k} \in \mathcal{V}^k\}$ contains at most $|\mathcal{V}|^k$ states. In the idealized continuous comparison, $\mathcal{A}_k^{\mathrm{soft}} = \mathbb{R}^{k\times d}$ contains this set. For any distortion D_θ defined on both sets,

$$\inf_{Z \in \mathcal{A}_k^{\mathrm{soft}}} D_\theta(H, Z) \le \min_{Z \in \mathcal{A}_k^{\mathrm{hard}}} D_\theta(H, Z). \tag{20}$$

This inclusion alone cannot establish a strict advantage: the acting model might assign the same next-token distribution to a reachable hard state and to the full sequence.

Let p_H be the full-context next-token distribution and $\delta_k(H) = \min_{Z \in \mathcal{A}_k^{\mathrm{hard}}} \|p_H - p_\theta^{>\ell^\star}(\cdot \mid Z)\|_1$. For any two categorical distributions p, q, Pinsker's inequality in natural-log units is

$$D_{\mathrm{KL}}(p\|q) \ge \tfrac{1}{2}\|p-q\|_1^2. \tag{21}$$

For completeness, take $A = \{i : p_i \geq q_i\}$, $a = p(A)$, and $b = q(A)$. Then $\|p - q\|_1 = 2(a - b)$. Grouping outcomes into A and its complement cannot increase KL, while the binary KL as a function of a has second derivative $1/[a(1-a)] \geq 4$ and vanishes with zero derivative at $a = b$. Hence binary KL is at least $2(a-b)^2$, which proves Equation 21. It follows that

$$\min_{Z \in \mathcal{A}_k^{\text{hard}}} D_\theta(H, Z) \geq \tfrac{1}{2}\delta_k(H)^2. \tag{22}$$

If $\delta_k(H) > 0$, no hard-token sequence of length k reproduces the full-context next-token distribution exactly. The finiteness of the hard output set makes this condition equivalent to p_H being absent from that set.

For ASTC's particular output Z_{ASTC}, the conditional logit bound in Equation 19 gives $D_\theta(H, Z_{\text{ASTC}}) \leq 2E$. Therefore

$$E < \delta_k(H)^2/4 \quad \Longrightarrow \quad D_\theta(H, Z_{\text{ASTC}}) < \min_{Z \in \mathcal{A}_k^{\text{hard}}} D_\theta(H, Z). \tag{23}$$

The statement compares equal physical length, fixed positions, and one next-token distribution. Its premise requires an actual bound on all downstream attention and recurrent defects. The ambient continuous set is larger than ASTC's attainable mean-pooled set, and finite-precision execution further restricts it. Thus the set inclusion establishes capacity, while Equation 23 specifies when one ASTC representation realizes a strict advantage.

For completeness, the pre-cast representative in Equation 5 is the least-squares solution. With $\bar{\mathbf{h}}_G = |G|^{-1} \sum_{i \in G} \mathbf{h}_i$, the identity

$$\sum_{i \in G} \|\mathbf{h}_i - \mathbf{z}\|_2^2 = \sum_{i \in G} \|\mathbf{h}_i - \bar{\mathbf{h}}_G\|_2^2 + |G| \|\mathbf{z} - \bar{\mathbf{h}}_G\|_2^2 \tag{24}$$

shows that the FP32 mean minimizes squared hidden-state error. This does not minimize downstream KL; ASTC checks reconstruction again after casting.

A.6 Proof of Theorem 4.1

Proof. Let p be the unchanged physical length of the cached prefix and m the number of newly processed positions before the commit, so $n_{\text{phys}} = p + m$ and $0 < r < m$. At one full causal attention layer, the new queries create

$$Q_{\text{before}} = pm + \frac{m(m+1)}{2} \tag{25}$$

query–key pairs: each attends the p cached keys and its causal prefix in the new block. After replacing r new positions, the unchanged cached prefix still contributes p keys, but only $m - r$ new queries remain. Thus

$$Q_{\text{after}} = p(m-r) + \frac{(m-r)(m-r+1)}{2}. \tag{26}$$

Subtracting gives the same-pass reduction

$$Q_{\text{before}} - Q_{\text{after}} = pr + \frac{r(2m-r+1)}{2} = \frac{r(2n_{\text{phys}} - r + 1)}{2}. \tag{27}$$

The fresh-prefill case is $p = 0$. Every later query that actually reaches this attention layer under the same fixed continuation has exactly r fewer historical keys while the compared caches retain their length difference; N_q such queries add rN_q saved pairs. Summing over the $A_>$ affected full-attention layers proves Equation 10. The group positions need not be contiguous because the count depends on the physical length after stable sorting, not on which original positions were merged. The result excludes work already spent on the cached prefix and assumes no intervening change to the compared cache that alters this r-position difference. A future token compressed before reaching these layers is not counted in N_q. □

A.7 Proof of Lemma 4.2

Proof. The query follows the greatest logical position of G, so all g original members are visible in the uncompressed run and their one representative is visible in the compressed run. Their identical scores give group unnormalized masses ge^s and e^s, respectively. Because the retained positions are unchanged, their attention masses are

$$A_R = \frac{Z_R}{Z_R + ge^s}, \qquad \widetilde{A}_R = \frac{Z_R}{Z_R + e^s}. \tag{28}$$

Direct subtraction yields

$$\widetilde{A}_R - A_R = \frac{Z_R(g-1)e^s}{(Z_R + e^s)(Z_R + ge^s)} > 0, \tag{29}$$

since $g > 1$ and $Z_R > 0$. The two attention outputs are

$$\mathbf{o} = A_R\overline{\mathbf{v}}_R + (1 - A_R)\mathbf{v}_G, \qquad \widetilde{\mathbf{o}} = \widetilde{A}_R\overline{\mathbf{v}}_R + (1 - \widetilde{A}_R)\mathbf{v}_G. \tag{30}$$

Subtracting and taking the inner product with any $\mathbf{u}$ gives Equation 11. Positivity of the projected change requires $\mathbf{u}^\top(\overline{\mathbf{v}}_R - \mathbf{v}_G) > 0$; it is not a claim about arbitrary nonlinear decisions. If scores or values differ, the exact identity in Equation 16 separates attention-mass shift from value error. □

B AGENT TRAJECTORY EXAMPLES

Figure 7 shows two AppWorld trajectories. ASTC retains exact fields as hard tokens and replaces selected history positions with soft tokens. The reported attention-pair reductions describe logical computation in these trajectories, not measured wall-clock speedups.

VENMO total requested in the last 10 days

Tool result (20 requests)

```
cutoff: 2023-05-23T00:00:00
#6131  amount=78.0  created_at=2023-06-01T12:47:07                    IN
    description="Makeup ..."  sender="Marcus Smith"
... 18 more requests ...
#4001  amount=10.0  created_at=2023-04-21T19:16:37                    OUT
    description="Baking Supplies ..."  sender="Travis Wilson"
```

Next model call: compressed history

hard tokens: cutoff, IDs, dates, amounts

618 selected positions -> [s1] ... [s16]

remaining unselected positions stay hard | 602 fewer positions

Agent filters 11 in-window requests and computes 833.

complete_task(answer=833.0) | Base 2/2 ASTC 2/2

24.06% fewer logical Attention pairs on the 14-action ASTC path

SPOTIFY update Road Trip Playlist #654

Tool results: playlist + roommates' messages

```
playlist: #116 Paper Planes; #295 City of Fallen Angels
Jose:  add => Where Shadows Come to Life / Olivia Roberts
       remove => Paper Planes / Seraphina Dawn
Chris: add => The Enigma of Love / Mia Sullivan
       remove => City of Fallen Angels / Hazel Winter
Paul:  No suggestions, looks good to me!
```

Next model call: compressed history

hard tokens: playlist #654 and numeric song IDs

160 newly selected positions -> [s1] ... [s16]

remaining unselected positions stay hard | 144 fewer positions

Agent finds #83/#152; adds them; removes #116/#295.

Updated playlist | Base 5/5 ASTC 5/5

19.10% fewer logical Attention pairs on the 23-action ASTC path

Figure 7: ASTC compression in Venmo and Spotify agent trajectories.

C Experimental Setup and Protocols

C.1 General Setup

Benchmarks. We evaluate ASTC on three long-horizon agentic benchmarks with distinct interaction structures. AppWorld provides a simulated ecosystem of everyday applications and APIs for interactive coding agents; we use its full 168-task Test-Normal split (Trivedi et al., 2024). OfficeBench covers realistic office-automation workflows that require planning, application switching, and grounded actions across productivity software (Wang et al., 2024). 8-Objective QA groups eight Natural Questions into each task and requires an agent to gather evidence with a Wikipedia search tool before returning all eight answers (Kwiatkowski et al., 2019; Zhou et al., 2026; Kang et al., 2026). Together, they test compression across API interaction, office automation, and multi-question information gathering.

Implementation. The three main benchmarks use the frozen Qwen3.5-9B acting model. For Table 1, all methods within each benchmark share the acting-model serving configuration and evaluation protocol. Base and ASTC also share the agent scaffold, prompts, decoding policy, context limit, and action budget.

Model checkpoints. The main experiments use `Qwen/Qwen3.5-9B` at Hugging Face revision `c202236235762e1c871ad0ccb60c8ee5ba337b9a` (Qwen Team, 2026). The transfer study also uses `moonshotai/Kimi-Linear-48B-A3B-Instruct` at revision `e1df551a447157d4658b573f9a695d57658590e9` (Moonshot AI, 2025).

Bounded pooling. Selected positions are divided into at most M groups in causal order. In each group, an existing member closest to the mean direction anchors an initial trim by relative ℓ_2 error. ASTC then averages the remaining states in FP32, casts the mean to the runtime dtype, and accepts a group of at least two members only if its maximum post-cast relative error is at most τ_h. Rejected and trimmed positions remain hard tokens. This check bounds hidden-state reconstruction, not downstream attention scores.

Baselines. Base is the uncompressed acting LLM. We compare ASTC with representative hard-token approaches, including prompted compression, LLMLingua (Jiang et al., 2023), LLMLingua-2 (Pan et al., 2024), and the UT and UTCO variants of ACON (Kang et al., 2026), as well as LCLM, a trained soft-token compressor (Li et al., 2026). These external-compressor comparisons are reported on AppWorld and OfficeBench; 8-Objective QA also includes Prompting, both LLMLingua variants, both ACON variants, and LCLM. The newly reported AppWorld LLMLingua-2 and OfficeBench Prompting and LLMLingua runs compress observations; the remaining hard-token baselines compress the full context, whereas ASTC compresses eligible tool observations in-model. All methods that require training use Qwen3.5-9B as the backbone and are evaluated with the same benchmark protocol. End-to-end workload includes acting-model requests, auxiliary compressor-service calls, compressor-generated tokens, and repeated prefills. The 8-Objective QA counts include the LLMLingua v1 and LLMLingua-2 service forwards.

Workload metrics. Decode KV counts query–cache interactions during autoregressive decoding. Attention Pairs also includes prefill and auxiliary model calls, while Generated Tokens counts all discrete outputs. These are analytical counts, not measured latency or GPU operations. Each workload metric in Table 1 is normalized to Base within its benchmark.

C.2 Benchmark-Specific Configurations

For the three main benchmarks in Table 1, we follow the benchmark tasks, agent scaffolds, model-serving settings, inference limits, and evaluation procedures of ACON (Kang et al., 2026). The benchmark settings and ASTC-specific compression policies are detailed below.

AppWorld. We evaluate all 168 AppWorld 0.1.3.post1 Test-Normal tasks under this ACON-aligned setup. The frozen Qwen3.5-9B agent uses greedy decoding, at most 50 actions, 768 output tokens per action, and a 32,768-token logical context. The model seed is 20260914, and task i uses seed $20260912 + i$ in the released task order. ASTC operates after zero-based GDN layer 10 and before attention layer 11, retaining at most 16 soft tokens per tool observation. Its fixed reader is calibrated from 22 records across 12 Train/Dev tasks, without weight updates. Each head uses leading eigen-

vectors of trace-normalized tool-end state covariances; the ridge and RMS stabilizers are $\lambda = 10^{-4}$ and $\epsilon = 10^{-6}$. The ordinary-data thresholds are $\chi < 0.60$, $\psi < 0.80$, and merge error ≤ 0.50; the conservative catalog thresholds are 0.55, 0.74, and 0.55, respectively. A commit must save at least 256 downstream token-layer positions and 5% of eligible tool positions. API contracts, catalogs, and exact-value fields are protected.

OfficeBench. We use the official Test split, Qwen3.5-9B, seed 20260920, and the official environment and evaluator. The agent uses at most 50 iterations, 768 output tokens per action, and a 32,768-token context. OfficeBench uses the same ASTC layer, selection thresholds, merge bound, and persistent cache as AppWorld. Only later byte-identical repetitions of an application-interface scaffold are eligible for compression; the first copy, changed scaffolds, observations, task instructions, system text, and prior actions remain explicit. Base and ASTC share the prompt and action budget.

8-Objective QA. All methods evaluate the same ACON Test tasks. The frozen Qwen3.5-9B acting model uses BF16 with vLLM 0.18.0, a 131,072-token model limit, engine seed 42, and at most 30 agent iterations for every method. ASTC compresses eligible tool content at the AppWorld layer boundary; the reported policy uses $(\chi, \psi) = (0.55, 0.75)$ for ordinary tool data. EM and F1 score the eight answers per task. All methods share the same serving and evaluation configuration, and workload is normalized to the corresponding Base run.

C.3 Additional-Dataset Protocols

RULER. We evaluate the same 32K inputs in Base and ASTC with the frozen Qwen3.5-9B model and a 32,768-token context limit. RULER scores answers by task-specific string matching and averages over its 13 task configurations. We use 20 examples per task, rather than the reference configuration's 500. Thus the reported accuracy is a 260-example aggregate; matching aggregate scores do not imply identical results on every task or at the larger default sample size. ASTC uses the Stage-3 data branch with $(\chi, \psi) = (0.60, 0.80)$ and merge-error threshold 0.50. Decode KV and Attention Pairs are analytical query–cache and causal-pair counts; Gen. is generated tokens, all normalized to Base.

LongBench v2. We use all 503 four-choice questions in the public `train` split (Bai et al., 2024). Base and ASTC share the frozen Qwen3.5-9B model and a 262,144-token context limit; examples that exceed the limit are truncated before either method processes them. Of the 503 examples, 100 require truncation. The inference configuration permits at most 16 new tokens; the recorded runs generate two tokens per example in both variants. ASTC uses the Stage-3 catalog branch with $(\chi, \psi) = (0.55, 0.74)$, merge-error threshold 0.55, and minimum net removal of one token. Overall and subgroup accuracy are exact-match rates for the extracted choice. Reported KV counts and cache bytes are analytical logical workload measures, not latency or measured peak device memory.

For MBPP, we use the sanitized task snapshot from Archer (He et al., 2026).

MBPP. We use its 427 sanitized MBPP problems (Austin et al., 2021). This interactive code-repair protocol is not numerically comparable with one-shot MBPP evaluations. The model submits a complete Python program, receives public-test feedback, and revises it; a third attempt is allowed if the second fails. The final program is scored once. Both variants use greedy decoding, at most 768 generated tokens per attempt, and a 32,768-token context limit. ASTC uses its Stage-3 calibration, thresholds, and 16-soft-token bound. Initial prompts, completed code, and public-test feedback are eligible for compression, while special control tokens remain explicit. Both variants re-prefill the full transcript before each turn; workload includes these repeated prefills and all decoding calls.

D Component-Ablation Details

Each variant in Table 2 changes one component while keeping the frozen model and inference configuration fixed. Hard deletion uses ASTC's selector but drops positions instead of emitting soft tokens.

Removing protection lowers accuracy from 33.33% to 20.83% (21 fewer successes), even as Attention Pairs fall to 68.63% of Base; Generated Tokens rise to 127.05%. Losing API contracts, identifiers, or exact values can invalidate later actions and lengthen recovery trajectories.

Without χ, accuracy returns to Base's 32.14%, two fewer successes than full ASTC. Attention Pairs fall to 78.11%, but Decode KV and Generated Tokens rise relative to full ASTC. Without ψ, accuracy is 32.74% and all three workload measures rise. The tests address distinct cases: new writes to recurrent state and values no longer readable at the endpoint.

Hard deletion loses one success and raises Decode KV, Attention Pairs, and Generated Tokens from ASTC's 81.55%, 80.70%, and 94.83% to 88.83%, 86.85%, and 99.58%, respectively. Retaining the group-level signal as a soft token supports later decisions in this comparison.

E Additional Dataset Experiments

We evaluate ASTC with the frozen Qwen3.5-9B model on RULER 32K (Hsieh et al., 2024), LongBench v2 (Bai et al., 2024), and interactive code repair on MBPP (Austin et al., 2021). The respective sample counts are 260, 503, and 427. All comparisons use a matched Base run within each benchmark; the protocols are detailed in Appendix C.3.

Table 3: Additional-dataset results.

	RULER 32K			**LongBench v2**			**MBPP**		
Method	Acc.% ↑	D-KV% ↓	Attn.% ↓	Acc.% ↑	D-KV% ↓	Cache TB ↓	Pass% ↑	D-KV% ↓	Attn.% ↓
Base	97.173	100.00	100.00	40.95	100.00	2.065	**84.78**	100.00	100.00
ASTC	97.173	**93.96**	**85.57**	40.95	**92.93**	**1.921**	84.07	**92.42**	**92.45**

On RULER, ASTC matches Base's aggregate answer score to the reported precision while reducing Decode KV by 6.04% and Attention Pairs by 14.43%. It migrates 1,045,409 of 6,485,833 eligible tokens into 1,468 soft tokens, for a 12.56% net prompt-token reduction. This 20-example-per-task run is smaller than RULER's 500-example reference configuration.

LongBench v2 yields 206/503 correct for both variants, with four Base-only and four ASTC-only successes. ASTC reduces Decode KV visits by 7.07%, all-layer final KV tokens by 6.19%, and analytical KV-cache bytes by 6.98%. The subgroup changes are mixed (Base/ASTC: Easy 44.79/45.83%, Hard 38.59/37.94%; Short 44.44/45.56%, Medium 40.00/39.53%, Long 37.04/36.11%). The 262,144-token limit truncates 100 examples in each variant; these figures do not establish a wall-clock speedup.

On MBPP, ASTC compresses at least one span in 182/427 trajectories and reduces workload; its paired success difference is -0.70 percentage points (95% bootstrap interval $[-1.87, 0.23]$). Mean generated tokens per problem are 89.2/89.4 on RULER, 2.0/2.0 on LongBench, and 337.4/329.0 on MBPP (Base/ASTC). The MBPP run does not establish a success-rate improvement.

F Extended Context Compression

We extend ASTC from tool observations to completed assistant history and the full context without retraining or changing the compression mechanism. This evaluates whether the same model-native criterion transfers across different context components.

Table 4: Effect of persistent compression scope on AppWorld.

Method	Acc. ↑	D-KV ↓	Attn. ↓	Gen. ↓
Base	32.14%	100.00%	100.00%	100.00%
Tool	**33.33%**	81.55%	**80.70%**	94.83%
Assistant	25.60%	**68.42%**	99.60%	**84.49%**
Full Context	32.14%	75.48%	98.63%	88.53%

Table 4 shows how the same training-free criterion behaves beyond tool observations. Tool-only compression improves accuracy while reducing all three workload measures. Compressing assistant history yields the largest Decode KV and generated-token savings but lowers accuracy; its nearly unchanged Attention Pairs indicate that extra prefill work offsets most of the decoding gain. Full-context compression matches Base accuracy and reduces Decode KV and generated tokens, with only a small Attention Pairs reduction. The criterion transfers across context components, but a broader scope does not necessarily reduce end-to-end attention work.

G CROSS-MODEL TRANSFER

We test whether ASTC transfers to the frozen Kimi-Linear hybrid backbone on AppWorld Test-Normal. Each model uses a reader calibrated for its own internal state and a matching uncompressed Base run. ASTC compresses eligible tool observations while keeping API contracts and authentication content explicit. Qwen3.5-9B uses Stage 3, while Kimi-Linear uses Stage 5. Within each model, Base and ASTC share task IDs and decoding settings; these settings need not match across models. Kimi-Linear covers the first 68 Test-Normal tasks in official order; Qwen3.5-9B covers all 168.

Table 5: AppWorld transfer from Qwen3.5-9B to Kimi-Linear.

Model	Method	Acc.% ↑	D-KV% ↓	Attn.% ↓	Gen.% ↓
Qwen3.5-9B	Base	32.14	100.00	100.00	100.00
	ASTC	33.33	81.55	80.70	94.83
Kimi-Linear-48B-A3B	Base	1.47	100.00	100.00	100.00
	ASTC	4.41	84.01	88.76	94.70

On Kimi-Linear, ASTC reduces Decode KV, Attention Pairs, and generated tokens. Successes rise from one to three on its 68-task subset. This single transfer test does not establish a general accuracy gain across backbones.