Learning to Evict: Policy-Driven KV Cache Optimization for Multi-Hop Multimodal Spatial Reasoning

Anonymous ACL submission

Abstract

Visualization-of-thought (VoT) has substantially advanced multimodal spatial reasoning by interleaving visual thinking with verbal reasoning. However, multi-hop VoT reasoning introduces a fundamental scalability bottleneck: the autoregressive accumulation of interleaved visual and textual tokens causes KV cache memory to grow rapidly across successive hops. We present ScoutKV, a framework that recasts KV cache eviction as a reinforcement learning problem. ScoutKV trains a dedicated lightweight policy agent per KV head, conditioned on key–value representations, cross-hop positional encodings, and modality embeddings, to predict each token's long-term utility across the full reasoning trajectory. Training combines offline supervised fine-tuning with future-query attention targets and online GRPO optimization with a multimodal process reward targeting low-entropy text anchors and modality-specific spatial anchors. ScoutKV consistently reduces FLOPs, latency, and peak memory while maintaining or improving accuracy across multimodal spatial reasoning benchmarks, and generalizes robustly to broader vision-language tasks.

1 Introduction

Multimodal large language models (MLLMs) have achieved strong performance on a wide range of perception and reasoning tasks by jointly processing visual and textual inputs. A particularly challenging class of problems is multimodal spatial reasoning (MSR), which requires understanding object layouts, movements, and spatial relations across complex environments. Recent work on visualization-of-thought (VoT) has shown that equipping MLLMs with explicit visual thinking — interleaving generated visual thoughts with verbal reasoning at each step — substantially improves performance on these tasks. Models such as MVoT and VoCoT demonstrate that multi-hop reasoning, where visual and textual thoughts are generated and accumulated across successive steps, can unlock capabilities that purely text-based chain-of-thought approaches cannot match.

However, multi-hop VoT reasoning introduces a fundamental scalability bottleneck. At each reasoning hop, the model generates both textual and visual tokens, all of which are appended to the growing context and cached in the key-value (KV) store. Since attention complexity scales quadratically with sequence length and the KV cache grows linearly with accumulated tokens, even moderate reasoning depths can exhaust the memory capacity of modern hardware accelerators.

KV cache eviction — selectively removing cached tokens to bound memory usage — is a natural remedy. Existing methods, however, are ill-suited to the multi-hop multimodal setting: prefill-stage approaches ignore cross-hop utility shifts, while heuristic eviction rules apply uniform criteria across all attention heads, failing to capture per-head behavioral diversity and leading to systematic eviction errors.

In this work, we cast KV cache eviction in multi-hop multimodal reasoning as a reinforcement learning problem and present ScoutKV, a framework that trains a dedicated lightweight policy agent for each KV attention head. Rather than relying on hand-crafted signals, each agent learns directly from data to score cached tokens for eviction, conditioned on key–value representations, cross-hop positional encodings, and modality embeddings. Training proceeds in two stages. In the first stage, a future-query supervised objective teaches the agents to predict which tokens will receive high attention from queries in subsequent hops, providing a direct proxy for cross-hop token relevance. In the second stage, we formulate eviction as a Markov Decision Process and apply GRPO with a multimodal process reward that

measures the impact of eviction on the generation quality of critical reasoning tokens — including low-entropy text anchors and modality-specific spatial anchors — aggregated over the full reasoning trajectory. Eviction sampling is modeled as a stochastic permutation policy, enabling efficient non-autoregressive exploration during both training and inference.

ScoutKV consistently reduces KV cache memory and computational overhead while maintaining or improving reasoning performance across a comprehensive suite of multimodal spatial reasoning benchmarks, and generalizes robustly to broader multimodal reasoning tasks. Our main contributions are:

- We introduce ScoutKV, a framework of per-head lightweight policy agents that score tokens for eviction conditioned on key–value representations, relative cross-hop positional encodings, and modality embeddings, trained via a two-stage pipeline combining future-query supervised fine-tuning and GRPO with multimodal process rewards.

- We propose a multimodal process reward that captures the impact of eviction decisions across the full reasoning trajectory, targeting both low-entropy text anchors and framework-specific spatial anchors.

- We demonstrate through extensive experiments that ScoutKV achieves strong efficiency–performance trade-offs across diverse spatial reasoning benchmarks and generalizes beyond its training distribution.

2 Related Work

2.1 Multimodal Spatial Reasoning

Driven by the introduction of Chain-of-Thought (CoT) prompting (Wei et al., 2022), Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs) (Brown et al., 2020; Touvron et al., 2023; Guo et al., 2025) have achieved substantial performance gains in complex reasoning paradigms. To tackle the inherent challenges of multimodal spatial reasoning, recent literature seeks to extend symbolic reasoning into the visual domain. Representative approaches, such as Visualization-of-Thought (VoT) (Wu et al., 2024b; Li et al., 2025b), interleave explicit visual thinking with verbal reasoning steps, yielding breakthroughs in understanding complex object layouts and navigation tasks. However, these multi-hop reasoning frameworks formulate the reasoning trajectory as the autoregressive accumulation of visual and textual tokens, inevitably triggering an explosive growth in context length (Yang et al., 2025; Hsieh et al., 2024). Because the computational and memory complexities of the Transformer architecture scale quadratically with sequence length (Sheng et al., 2023; Dao et al., 2022), the exorbitant arithmetic overhead and memory bottlenecks fundamentally limit the scalability of MLLMs in deep spatial reasoning.

2.2 Cache Management and Token Pruning

The primary bottleneck in processing extended reasoning trajectories lies in the linear expansion of the Key-Value (KV) cache. To alleviate memory constraints, existing KV cache management techniques focus on low-bit quantization (Dettmers et al., 2022; Xiao et al., 2023a) and cache eviction. Eviction-based methods exploit the inherent sparsity of attention mechanisms to permanently discard low-utility tokens based on heuristic rules. For instance, several approaches rely on local sliding windows and attention sinks (Xiao et al., 2023b), or utilize accumulated attention scores and feature similarities as hard truncation criteria (Zhang et al., 2023; Adnan et al., 2024); others leverage the current query to inform eviction boundaries (Li et al., 2024; Tang et al., 2024). Although recent adaptive compression techniques based on offline training(Ge et al., 2023; Deng et al., 2025; Nawrot et al., 2024) have been proposed, they are overwhelmingly designed for unimodal text generation and typically execute irreversible eviction decisions after a single forward pass (Cai et al., 2025). Beyond cache-level eviction, directly pruning tokens at the sequence level serves as another effective trajectory for reducing computational complexity. In the textual domain, existing compression methods primarily involve discrete token reduction—via prompt engineering (Aytes et al., 2025) and instruction tuning (Liu et al., 2024)—or implicit reasoning approaches that project intermediate steps into continuous latent spaces (Hao et al., 2024; Cheng and Van Durme, 2024; Shen et al., 2025). Regarding visual token compression, predominant algorithms capitalize on the spatial redundancy of images by identifying and discarding uninformative visual patches to accelerate inference (Chen

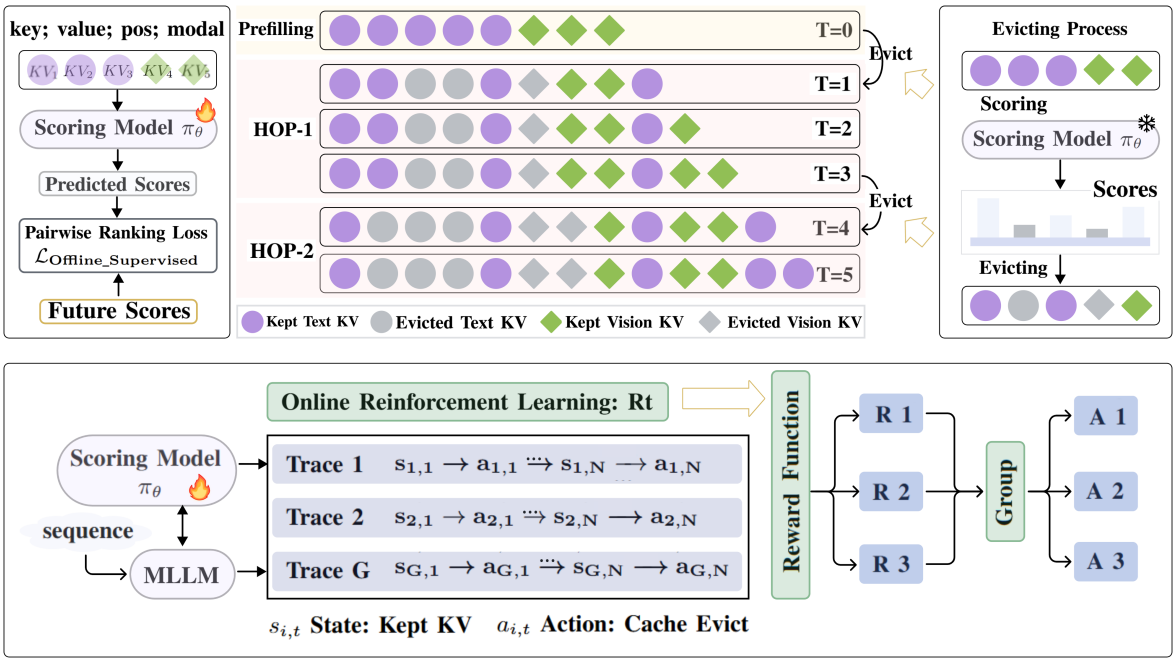

Figure 1: Overview of ScoutKV. The upper panel illustrates the offline supervised fine-tuning stage: at the end of each reasoning hop, the scoring agent π_θ assigns scores to cached text and visual tokens and triggers eviction; the agent is trained via Pairwise Ranking Loss using normalized future attention scores as supervision targets. The lower panel illustrates the online reinforcement learning stage: G independent eviction traces are sampled from the same input sequence, evaluated by the multimodal process reward R, and used to compute group-normalized advantages for GRPO-based policy optimization.

et al., 2024a,b; Tan et al., 2025; Zhang et al., 2024). In multi-hop multimodal reasoning, the semantic utility of tokens undergoes drastic and dynamic shifts across consecutive hops. Consequently, these static or inter-hop-agnostic heuristics are prone to suboptimal pruning, leading to the irrecoverable loss of critical spatial anchors.

2.3 Reinforcement Learning Optimization

GRPO (Shao et al., 2024) has gained wide adoption for training language models with non-differentiable rewards by normalizing advantages within a group of sampled outputs. DeepSeek-R1 (Guo et al., 2025) demonstrated that pure reinforcement learning from verifiable rewards can produce strong long-chain reasoning without supervised fine-tuning. Process reward models (Lightman et al., 2023) provide denser supervision by scoring intermediate reasoning steps rather than only final outcomes, offering better credit assignment for multi-step tasks. The Plackett–Luce model (Plackett, 1975) and Gumbel-Sort sampling (Kool et al., 2019) provide principled stochastic relaxations of ranking decisions that allow gradient-free or gradient-based optimization over permutations. MInference (Jiang et al., 2024) demonstrated that attention pattern diversity across heads can be exploited for efficient sparse attention computation, independently motivating our per-head agent design.

3 Methodology

Our framework is illustrated in Figure 1 and proceeds in three stages. First, in the multi-hop dynamic eviction formulation, we define a dedicated lightweight scoring agent per KV head, which scores and evicts cached tokens at the end of each reasoning hop. Second, in the supervised fine-tuning stage, the agents are trained offline via Pairwise Ranking Loss, using the normalized average future attention score as the supervision target to instill long-term utility prediction. Third, in the reinforcement learning stage, we apply GRPO with a multimodal process reward to optimize the policy online, correcting the distribution shift introduced by offline training.

3.1 Multi-Hop Dynamic Eviction Formulation

We partition the multimodal reasoning sequence into T reasoning hops, denoted $\mathcal{S} = \{H_1, H_2, \ldots, H_T\}$, where each hop $H_t = [X_t^{\text{text}}; X_t^{\text{vis}}]$ contains the interleaved text and visual tokens generated at that step. As reasoning proceeds, tokens from both modalities accumulate in the KV cache, and eviction is triggered at the end of each hop.

For layer l and KV head h, we maintain a dedicated lightweight scoring agent $\pi_\theta^{l,h}$. In models with Grouped Query Attention (GQA), each KV head is shared by a group of query heads; defining one agent per KV head therefore aligns with the physical granularity of cache allocation. At the end of hop t, the input feature for the i-th cached token is:

$$\varphi_{i,t}^{l,h} = \left[k_i^{l,h};\ v_i^{l,h};\ \mathbf{p}_{i,t};\ e(m_i)\right] \tag{1}$$

where $k_i^{l,h}$ and $v_i^{l,h}$ are the key and value vectors at the corresponding layer and KV head; $m_i \in \{\text{text}, \text{vis}\}$ is the modality label, and $e(\cdot)$ is a learnable modality embedding table of dimension $d_e = 8$. Rather than imposing hand-crafted rules, this embedding enables the agent to distinguish modality-specific distributional differences and learn cross-modal eviction preferences from data. The cross-hop positional encoding $\mathbf{p}_{i,t}$ is defined as:

$$\mathbf{p}_{i,t} = \text{Sinusoidal}(\Delta\text{hop}_{i,t}) \in \mathbb{R}^{d_p} \tag{2}$$

where $\Delta\text{hop}_{i,t} = t - \text{hop}(i)$ is the hop distance between the current eviction step and the hop in which token i was generated, and $d_p = 8$. Mapping this scalar distance into a low-dimensional continuous space preserves extrapolation to unseen hop distances while preventing the signal from being drowned out when concatenated with high-dimensional KV vectors.

The agent produces a score for each token:

$$s_{i,t}^{l,h} = \pi_\theta^{l,h}\left(\varphi_{i,t}^{l,h}\right) \tag{3}$$

Each scoring agent $\pi_\theta^{l,h}$ is implemented as a two-layer MLP: the input feature $\varphi_{i,t}^{l,h}$ is first normalized by a LayerNorm layer, then passed through a fully connected layer with hidden dimension 256, followed by a linear projection to a scalar output. The parameters of each agent are trained independently without weight sharing, allowing each head's eviction policy to adapt to its own distinct attention distribution.

Eviction Sampling. Deterministic top-$(1-\rho)$ truncation eliminates the exploration needed for policy gradient optimization and prevents generating diverse eviction traces from the same input. We therefore model score-based token ranking as a stochastic permutation policy following the Plackett–Luce model, whose joint probability over N cached tokens is:

$$\pi_\theta(\sigma \mid \varphi_{1,t}^{l,h}, \ldots, \varphi_{N,t}^{l,h}) = \prod_{i=1}^{N} \frac{\exp\left(s_{\sigma_i,t}^{l,h}\right)}{\sum_{j=i}^{N} \exp\left(s_{\sigma_j,t}^{l,h}\right)} \tag{4}$$

We exploit the Gumbel-Sort trick for efficient sampling: drawing i.i.d. noise $g_i \sim \text{Gumbel}(0, 1)$ and sorting the perturbed scores:

$$\sigma = \text{argsort}_{i\in[N]}\left(s_{i,t}^{l,h} + g_i\right) \tag{5}$$

This procedure is fully non-autoregressive, requiring only a single agent forward pass followed by a sort, and is thus easily parallelized on modern hardware. The top-$(1-\rho)$ fraction of tokens under σ are retained as the KV cache for hop $t+1$.

3.2 Future-Query Supervised Fine-Tuning

The SFT stage trains the scoring agents to predict the long-term utility of cached tokens. For the eviction step at hop t, we collect the query vectors of all text tokens from subsequent hops to form the future query set:

$$Q_{\text{future}}^t = \{q_j \mid j \in H_{>t}^{\text{text}}\} \tag{6}$$

We restrict Q_{future}^t to text queries for two reasons. First, in VoT reasoning, text tokens carry the logical backbone of the multi-hop process—governing spatial relation descriptions, step-to-step chaining, and final answer formation—and thus serve as the primary determinants of reasoning quality. Second, within a single hop the visual generation and its accompanying textual narration are semantically aligned, so text queries already capture cross-modal dependencies and provide a reliable proxy for the long-term utility of *both* visual and textual tokens. Visual tokens, by contrast, are numerous and contain substantial spatial redundancy; incorporating their query vectors would

introduce noise that degrades the quality of the supervision targets. The supervision target for the i-th KV pair is the normalized average attention received from future text queries:

$$\alpha_{i,t}^{l,h} = \frac{1}{|Q_{\text{future}}^t|} \sum_{j \in Q_{\text{future}}^t} \text{Attention}(q_j,\, k_i^{l,h}) \quad (7)$$

where $\text{Attention}(q_j, k_i^{l,h})$ denotes the softmax-normalized attention weight. Averaging over $|Q_{\text{future}}^t|$ compensates for the fact that hops closer to the start of reasoning have substantially more future queries than later hops; without this normalization, target scores would decay systematically with t, making cross-hop ranking comparisons unreliable.

We train the agents with a Pairwise Ranking Loss:

$$\mathcal{L}_{\text{SFT}}(\theta) = \sum_t \sum_{\substack{(i,j):\\ \alpha_{i,t} > \alpha_{j,t}}} \max(0,\, \gamma - (s_{i,t} - s_{j,t})) \quad (8)$$

where γ is a margin constant, and $s_{i,t}$ abbreviates $s_{i,t}^{l,h}$ where the layer and head indices are clear from context.

3.3 Reinforcement Learning with Multimodal Process Rewards

Offline SFT constructs supervision signals from static traces, yet eviction at inference time continuously alters the hidden-state distribution of subsequent hops, introducing a distribution shift that cannot be corrected by offline training alone. To address this, we formulate eviction as a Markov Decision Process (MDP) and apply GRPO for online policy optimization. At each hop t, the state s_t is the retained KV set, and the action a_t is the eviction index set sampled under π_θ; the reward R is defined below.

Let $L_{\text{full}}(k)$ and $L_{\text{evict}}(k)$ denote the cross-entropy loss of the model on token k under the full KV cache and after eviction, respectively. The loss increment is $\Delta L_t(k) = L_{\text{evict}}(k) - L_{\text{full}}(k)$. The reward aggregates over a set of critical tokens spanning the full generated sequence, making it a sequence-level signal.

The critical token set $\mathcal{K}_t = \mathcal{K}_t^{\text{text}} \cup \mathcal{K}_t^{\text{spatial}}$ consists of two components:

- **Low-entropy text anchors** $\mathcal{K}_t^{\text{text}}$: text tokens whose generation entropy falls in the lower end of the sequence distribution. These tokens carry high prediction confidence and serve as semantic consistency constraints; even small loss increments at these positions tend to propagate and degrade downstream reasoning quality.
- **Spatial anchors** $\mathcal{K}_t^{\text{spatial}}$: defined differently per framework. Under Volcano, these are the bounding-box coordinate tokens that anchor visual grounding. Under Anole, these are *Text-Conditioned Visual Anchors* — the top-K visual tokens in hop t that receive the highest normalized average attention from the text queries in the same hop:

$$\mathcal{K}_t^{\text{Anole}} = \underset{i \in H_t^{\text{vis}}}{\text{Top-K}} \frac{1}{|H_t^{\text{text}}|} \sum_{j \in H_t^{\text{text}}} \text{Attention}(q_j, k_i) \quad (9)$$

The sequence-level process reward is then:

$$R = -\sum_{t=1}^{T} \sum_{k \in \mathcal{K}_t} [\Delta L_t(k)]^2 \quad (10)$$

Policy Optimization. For a given input, we draw G independent sets of Gumbel noise under the old policy $\pi_{\theta_{\text{old}}}$ to produce G distinct eviction traces. Each trace reflects a different eviction sampling outcome without requiring separate autoregressive generation.

We distinguish the cost of trace sampling from the cost of reward evaluation. Sampling G eviction traces only requires G agent forward passes with Gumbel-Sort, and does not involve additional autoregressive decoding.

To compute per-trace rewards, we replay the frozen backbone under each sampled eviction trajectory with teacher forcing, and measure L_{evict} on the critical tokens, while L_{full} is computed once from the full-cache reference run. This process only requires forward passes of the backbone without any parameter updates.

Therefore, the RL optimization introduces moderate overhead from forward-only replay, while avoiding expensive autoregressive generation and keeping the training efficient in practice. Per-trace rewards are normalized within the group of G traces sampled from the same input sequence to estimate advantages:

$$\hat{A} = \frac{R - \text{Mean}(R)}{\text{Std}(R)} \quad (11)$$

The agents are then optimized by maximizing the GRPO objective:

$$\mathcal{J}(\theta) = \mathbb{E}\Bigg[\sum_t \min\Big(r_t(\theta)\hat{A}, \quad \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}\Big) - \beta \mathcal{D}_{\text{KL}}(\pi_\theta \parallel \pi_{\text{ref}})\Bigg] \tag{12}$$

where $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{\text{old}}}(a_t|s_t)$ is the importance sampling ratio, ϵ is the clipping threshold, β controls KL regularization, and π_{ref} is the SFT-stage reference policy, included to prevent the updated policy from drifting too far from its initialization.

4 Experiments

4.1 Experimental Setup

Models and Implementation. We evaluate ScoutKV on two multimodal interleaved reasoning architectures. (1) ScoutKV is integrated into Volcano, which interleaves textual tokens with RefBind-based visual tokens for fine-grained grounding. We use the VoCoT-dataset to train the per-head KV scoring agents. (2) ScoutKV is integrated into Anole-7B, a model that generates mental images to support abstract spatial reasoning. We use the MVoT-dataset to train the per-head KV scoring agents. Importantly, ScoutKV requires no fine-tuning of the backbone model — only the lightweight scoring agent associated with each KV head is trained, while all backbone parameters remain frozen throughout. All experiments are conducted on RTX A6000 GPUs. In this experiment, we set the GRPO sample group size to $G = 8$ and the pairwise ranking margin to $\gamma = 0.1$. To constrain the policy updates, the GRPO clipping threshold is configured to $\epsilon = 0.2$. Additionally, the KV cache retention ratio is explicitly set to $1 - \rho = 60\%$.

Baselines. We compare against the following baselines: **SoT**, a static prompt-based token budget method; **LightFastV**, a modular heuristic baseline combining LightThinker for text compression and FastV for visual pruning; **SparseVLM**, which iteratively sparsifies visual tokens via learned importance scores; **Heima**, a latent-space reasoning baseline that compresses intermediate reasoning steps into continuous vectors.

Metrics. We report accuracy (Acc.↑), computational cost (FLOPs↓), and inference latency (Lat.↓) across all benchmarks, with memory usage (Mem.↓) additionally reported for static spatial reasoning and general VQA benchmarks.

4.2 Dataset Details

We evaluate ScoutKV on three categories of benchmarks to comprehensively assess spatial reasoning capability, general vision-language understanding, and hallucination resistance.

(1) Static Spatial Reasoning Benchmarks. We evaluate on four tasks: **VSR** is a binary classification task in which the model judges whether a textual claim is visually supported, evaluated on the unseen zero-shot test split (1,222 samples); **Emb-Spatial** focuses on embodied spatial understanding with fine-grained relational queries (3,625 samples); **V-Star** tests visual search and fine-grained object localization in cluttered scenes (238 samples); **Winoground** evaluates visio-linguistic compositionality by requiring the model to match image–caption pairs that differ in subtle semantics (800 samples).

(2) Dynamic Spatial Reasoning Benchmarks. Three multi-step visual reasoning tasks over simulated environments are included: **MAZE** consists of procedurally generated 2D grid mazes requiring long-horizon spatial path planning (1,255 test samples); **MiniBehavior** is derived from a printer-installation simulation, testing procedural task planning across grid environments ranging from 7×7 to 10×10 (1,604 test samples); **FrozenLake** is adapted from OpenAI Gym's FrozenLake environment, requiring navigation under uncertainty with sparse rewards (1,664 test samples).

(3) General VQA and Hallucination Benchmarks. **GQA** uses the `testdev_balanced` split to evaluate compositional question answering (12,578 samples); **MMBench** covers 12 skill dimensions on the DEV split (4,329 samples); **POPE** probes object hallucination via binary yes-or-no questions on the adversarial subset (3,000 samples); **AMBER** evaluates open-ended generation using the CHAIR hallucination metric (1,004 samples).

4.3 Main Results

Tables 1, 2, and 3 summarize the full evaluation of ScoutKV across three benchmark categories. Throughout, memory figures reflect runtime GPU peak memory; FLOPs are an analytical arithmetic count while latency is measured end-to-end, so the two are not strictly co-monotonic across methods.

Table 1: Comparison on multimodal spatial reasoning (split into two blocks for readability).

(A) VSR and V-Star.

Methods	VSR				V-Star			
	Acc.(%)↑	FLOPs(G)↓	Lat.(s)↓	Mem.(GB)↓	Acc.(%)↑	FLOPs(G)↓	Lat.(s)↓	Mem.(GB)↓
Volcano	67.18	19842.37	0.63	8.91	58.40	21785.41	0.69	9.37
SoT	53.22	17240.58	0.58	8.32	37.12	19820.16	0.64	8.94
LightFastV	57.19	15880.31	0.54	7.85	45.68	18340.33	0.59	8.43
Heima	50.42	**10138.96**	**0.39**	6.14	40.67	**13963.21**	**0.41**	6.42
SparseVLM	62.31	14827.25	0.49	7.23	55.95	17365.19	0.53	8.17
ScoutKV	**68.50**	13280.08	0.46	**6.02**	**60.85**	14620.33	0.50	**6.38**

(B) EmbSpatial and Winoground.

Methods	EmbSpatial				Winoground			
	Acc.(%)↑	FLOPs(G)↓	Lat.(s)↓	Mem.(GB)↓	Acc.(%)↑	FLOPs(G)↓	Lat.(s)↓	Mem.(GB)↓
Volcano	58.29	26542.87	0.72	9.73	68.37	27411.36	0.78	9.84
SoT	54.76	24175.63	0.66	9.18	62.90	25984.27	0.72	9.51
LightFastV	56.89	22904.75	0.62	8.71	64.21	24632.88	0.67	8.97
Heima	53.32	**17174.83**	**0.41**	6.92	64.03	**17811.47**	**0.49**	6.81
SparseVLM	62.32	20275.21	0.56	8.21	64.39	23155.26	0.63	8.35
ScoutKV	**62.47**	17852.47	0.52	**6.82**	**69.27**	18426.09	0.56	**6.79**

Table 2: Comparison on mental image generation tasks using Anole-7B across three dynamic reasoning benchmarks. We report accuracy (Acc.), compute cost (FLOPs), and inference latency (Lat.).ScoutKV provides a strong accuracy–efficiency trade-off. All methods are evaluated on the Anole-7B backbone.

Methods	MAZE			MINIBEHAVIOR			FROZENLAKE		
	Acc.(%)↑	FLOPs(G)↓	Lat.(s)↓	Acc.(%)↑	FLOPs(G)↓	Lat.(s)↓	Acc.(%)↑	FLOPs(G)↓	Lat.(s)↓
VoT	86.56	25640.14	0.79	64.40	24815.83	0.75	80.21	23792.67	0.74
MVoT	92.95	22130.56	0.71	95.14	21247.29	0.68	85.60	20591.34	0.67
Heima	80.37	**13243.92**	**0.42**	68.49	**10628.11**	0.47	79.32	**10870.96**	**0.38**
SparseVLM	87.26	17834.21	0.62	88.01	17223.23	0.56	80.29	17779.21	0.61
ScoutKV	**93.68**	14350.52	0.47	**95.72**	13680.51	**0.44**	**86.47**	13248.06	0.40

Static Spatial Reasoning (Table 1). On the Volcano backbone, ScoutKV achieves the highest accuracy on all four benchmarks. On VSR, ScoutKV reaches 68.50%, surpassing even the full-cache Volcano baseline (67.18%), indicating that principled eviction can actively sharpen spatial anchor signals rather than merely approximating full-cache behavior. ScoutKV also leads on V-Star (60.85%) and Winoground (69.27%), and matches SparseVLM on EmbSpatial (62.47% vs. 62.32%) while using 1.39 GB less memory. Relative to Volcano, ScoutKV reduces FLOPs by 32–33%, latency by 27–28%, and peak memory by 30–32%. Heima achieves lower FLOPs by collapsing reasoning steps into latent vectors, but this causes irreversible spatial information loss, lagging ScoutKV by 18.08 and 20.18 points on VSR and V-Star respectively. SoT and LightFastV suffer comparable accuracy drops from their head-uniform heuristics; SparseVLM recovers closer to full-cache accuracy but incurs consistently higher FLOPs and memory.

Dynamic Spatial Reasoning (Table 2). On Anole-7B, ScoutKV achieves the highest accuracy on MAZE (93.68%), MiniBehavior (95.72%), and FrozenLake (86.47%), even surpassing uncompressed MVoT on all three tasks—confirming that targeted multi-hop eviction can yield net quality gains, not merely an efficiency–accuracy trade-off. Relative to MVoT, FLOPs and latency are reduced by 35–36% and 34–40%. Heima again records the lowest compute cost but suffers severe accuracy degradation (e.g., 68.49% vs. 95.72% on MiniBehavior), as compact latent encoding destroys the intermediate spatial representations that successive reasoning hops depend on.

Table 3: Comparison on general QA and hallucination benchmarks (split into two blocks for readability). ↑ higher is better; ↓ lower is better (AMBER uses AMB↓).All methods are evaluated on the Volcano backbone.

(A) General VQA: GQA and MMBench.

Methods	GQA				MMBench			
	Acc.(%)↑	FLOPs(G)↓	Lat.(s)↓	Mem.(GB)↓	Acc.(%)↑	FLOPs(G)↓	Lat.(s)↓	Mem.(GB)↓
Volcano	64.40	6052.37	0.30	6.13	68.11	7285.41	0.35	6.52
SoT	60.23	5240.58	0.28	5.76	63.27	6720.16	0.32	6.08
LightFastV	61.23	4980.31	0.26	5.28	63.84	6240.33	0.29	5.59
Heima	63.42	**3120.36**	**0.20**	3.51	66.45	**4375.63**	**0.22**	4.12
SparseVLM	61.09	5021.36	0.29	5.91	63.77	6492.73	0.31	5.87
ScoutKV	**65.28**	4183.54	0.22	**3.47**	**68.69**	5021.62	0.25	**4.11**

(B) Hallucination: POPE and AMBER.

Methods	POPE				AMBER			
	Acc.(%)↑	FLOPs(G)↓	Lat.(s)↓	Mem.(GB)↓	AMB↓	FLOPs(G)↓	Lat.(s)↓	Mem.(GB)↓
Volcano	86.50	13285.70	0.43	6.21	4.60	14320.36	0.51	6.89
SoT	74.32	11598.12	0.39	5.82	6.13	13480.92	0.47	6.19
LightFastV	77.89	10492.24	0.36	5.38	5.42	11865.18	0.42	5.71
Heima	71.38	**6920.18**	**0.25**	**3.92**	5.01	**8046.73**	**0.29**	4.28
SparseVLM	78.32	11247.22	0.39	5.44	5.41	11764.31	0.40	5.74
ScoutKV	**87.65**	8927.56	0.31	3.96	**4.48**	9125.77	0.35	**4.01**

General VQA and Hallucination (Table 3). ScoutKV generalizes well beyond its training distribution. It ranks first on GQA (65.28%) and MMBench (68.69%) with the lowest memory (3.47 GB and 4.11 GB), reducing Volcano's FLOPs by 31–36%, latency by 27–31%, and peak memory by 36–43%. On hallucination benchmarks, ScoutKV exceeds the full-cache Volcano on POPE (87.65% vs. 86.50%) and achieves the best AMBER score (4.48), suggesting its eviction policy learns to discard tokens that contribute noise rather than grounding information. Methods relying on uniform heuristics (SoT, LightFastV) show systematic accuracy drops even on these out-of-distribution tasks.

Overall. Across all eleven benchmarks spanning static spatial reasoning, dynamic navigation, and open-ended VQA, ScoutKV consistently occupies the Pareto frontier in accuracy and efficiency. The uniformity of this pattern across substantially different task types confirms that per-head policy agents—conditioned on modality embeddings and cross-hop positional encodings—learn genuinely transferable token utility judgments rather than task-specific heuristics.

5 Conclusion

In this work, we presented ScoutKV, a reinforcement learning framework for efficient KV cache eviction in multi-hop multimodal spatial reasoning. By training a lightweight, per-head policy agent that leverages key–value representations, cross-hop positional encoding, and modality embeddings, ScoutKV moves beyond heuristic compression and static pruning. A two-stage training pipeline—supervised fine-tuning with future-query attention targets followed by GRPO optimization with a multimodal process reward—enables the agents to learn which tokens to retain for long-term reasoning utility. Extensive experiments on spatial reasoning, dynamic mental image generation, and general VQA benchmarks show that ScoutKV consistently reduces FLOPs, latency, and memory footprint while preserving or even improving task accuracy. The per-head design also reveals interpretable, head-specific eviction behaviors that static rules cannot capture. These results demonstrate that policy-driven, data-aware KV cache eviction offers a scalable and generalizable solution for efficient multimodal reasoning, paving the way for deeper, more resource-friendly visual-language models.

Limitations

ScoutKV trains a dedicated lightweight agent for each KV head of a specific backbone; although each agent is compact and the backbone remains frozen throughout, the agents must be retrained when transferred to a new model architecture. Additionally, the token retention ratio ρ is treated as a fixed global hyperparameter during both training and inference, whereas different tasks or varying input complexities may benefit from dynamic, input-adaptive budget allocation.

References

Muhammad Adnan, Akhil Arunkumar, Gaurav Jain, Prashant J Nair, Ilya Soloveychik, and Purushotham Kamath. 2024. Keyformer: Kv cache reduction through key tokens selection for efficient generative inference. *Proceedings of Machine Learning and Systems*, 6:114–127.

Simon A Aytes, Jinheon Baek, and Sung Ju Hwang. 2025. Sketch-of-thought: Efficient llm reasoning with adaptive cognitive-inspired sketching. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24307–24331.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Zefan Cai, Wen Xiao, Hanshi Sun, Cheng Luo, Yikai Zhang, Ke Wan, Yucheng Li, Yeyang Zhou, Li-Wen Chang, Jiuxiang Gu, and 1 others. 2025. R-kv: Redundancy-aware kv cache compression for training-free reasoning models acceleration. *arXiv e-prints*, pages arXiv–2505.

Jieneng Chen, Luoxin Ye, Ju He, Zhao-Yang Wang, Daniel Khashabi, and Alan Yuille. 2024a. Efficient large multi-modal models via visual context compression. *Advances in neural information processing systems*, 37:73986–74007.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024b. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer.

Jeffrey Cheng and Benjamin Van Durme. 2024. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359.

Chenlong Deng, Zhisong Zhang, Kelong Mao, Shuaiyi Li, Tianqing Fang, Hongming Zhang, Haitao Mi, Dong Yu, and Zhicheng Dou. 2025. Unigist: Towards general and hardware-aligned sequence-level long context compression. *arXiv preprint arXiv:2509.15763*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in neural information processing systems*, 35:30318–30332.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. 2023. Model tells you what to discard: Adaptive kv cache compression for llms. *arXiv preprint arXiv:2310.01801*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*.

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

Michael Igorevich Ivanitskiy, Rusheb Shah, Alex F Spies, Tilman Räuker, Dan Valentine, Can Rager, Lucia Quirke, Chris Mathwin, Guillaume Corlouer, Cecilia Diniz Behn, and 1 others. 2023. A configurable library for generating and manipulating maze datasets. *arXiv preprint arXiv:2309.10498*.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H Abdi, Dongsheng Li, Chin-Yew Lin, and 1 others. 2024. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. *Advances in Neural Information Processing Systems*, 37:52481–52515.

Emily Jin, Jiaheng Hu, Zhuoyi Huang, Ruohan Zhang, Jiajun Wu, Li Fei-Fei, and Roberto Martín-Martín. 2023. Mini-behavior: A procedurally generated benchmark for long-horizon decision-making in embodied ai. *arXiv preprint arXiv:2310.01824*.

Wouter Kool, Herke Van Hoof, and Max Welling. 2019. Stochastic beams and where to find them: The

gumbel-top-k trick for sampling sequences without replacement. In *International conference on machine learning*, pages 3499–3508. PMLR.

Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. 2025a. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970.

Zejun Li, Ruipu Luo, Jiwen Zhang, Minghui Qiu, Xuan-Jing Huang, and Zhongyu Wei. 2025b. Vocot: Unleashing visually grounded multi-step reasoning in large multi-modal models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3769–3798.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The twelfth international conference on learning representations*.

Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Cheng Jiayang, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2024. Can language models learn to skip steps? *Advances in Neural Information Processing Systems*, 37:45359–45385.

Jesse Mu, Xiang Li, and Noah Goodman. 2023. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36:19327–19352.

Piotr Nawrot, Adrian Łańcucki, Marcin Chochowski, David Tarjan, and Edoardo M Ponti. 2024. Dynamic memory compression: Retrofitting llms for accelerated inference. *arXiv preprint arXiv:2403.09636*.

Robin L Plackett. 1975. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202.

Zexuan Qiu, Zijing Ou, Bin Wu, Jingjing Li, Aiwei Liu, and Irwin King. 2025. Entropy-based decoding for retrieval-augmented large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4616–4627.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Xuan Shen, Yizhou Wang, Xiangxi Shi, Yanzhi Wang, Pu Zhao, and Jiuxiang Gu. 2025. Efficient reasoning with hidden thinking. *arXiv preprint arXiv:2501.19201*.

Ying Sheng, Lianmin Zheng, Binhang Yuan, Zhuohan Li, Max Ryabinin, Beidi Chen, Percy Liang, Christopher Ré, Ion Stoica, and Ce Zhang. 2023. Flexgen: High-throughput generative inference of large language models with a single gpu. In *International Conference on Machine Learning*, pages 31094–31116. PMLR.

Xudong Tan, Peng Ye, Chongjun Tu, Jianjian Cao, Yaoxin Yang, Lin Zhang, Dongzhan Zhou, and Tao Chen. 2025. Tokencarve: Information-preserving visual token compression in multimodal large language models. *arXiv preprint arXiv:2503.10501*.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. 2024. Quest: Query-aware sparsity for efficient long-context llm inference. *arXiv preprint arXiv:2406.10774*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: open foundation and fine-tuned chat models. arxiv. *arXiv preprint arXiv:2307.09288*, 10.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, and 1 others. 2025. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Qiucheng Wu, Handong Zhao, Michael Saxon, Trung Bui, William Yang Wang, Yang Zhang, and Shiyu Chang. 2024a. Vsp: Assessing the dual challenges of perception and reasoning in spatial planning tasks for vlms. *arXiv preprint arXiv:2407.01863*.

Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. 2024b. Mind's eye of llms: visualization-of-thought elicits spatial reasoning in large language models. *Advances in Neural Information Processing Systems*, 37:90277–90317.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023a. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, pages 38087–38099. PMLR.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023b. Efficient streaming language models with attention sinks, 2024. *URL https://arxiv. org/abs/2309.17453*, page 1.

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

Jintian Zhang, Yuqi Zhu, Mengshu Sun, Yujie Luo, Shuofei Qiao, Lun Du, Da Zheng, Huajun Chen, and Ningyu Zhang. 2025. Lightthinker: Thinking step-by-step compression. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13318–13339.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and 1 others. 2024. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, and 1 others. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.

Use of Large Language Models

This article was written with the assistance of Large Language Models (LLMs) for text polishing and expression optimization to improve readability and fluency. All core arguments, ideas, and content are original work by the author(s).

Appendix

A Baseline Methods Description

A.1 Reasoning Paradigm Baselines

SoT, proposed by Aytes et al. (Aytes et al., 2025), is a training-free prompt-based token budget compression method. Its core idea is to prepend a task-specific instruction (`<sketch>`) to the model's input prompt, compressing the reasoning process into cognitively inspired "sketch" phrases, thereby enforcing a predefined token retention ratio without modifying any model parameters. Since SoT is entirely based on prompt engineering, it requires no fine-tuning of the backbone or any auxiliary modules.

LightFastV is a heuristic baseline method that combines two independently designed compression components: LightThinker (Zhang et al., 2025) for textual token pruning and FastV (Chen et al., 2024b) for visual token pruning. LightThinker is implemented as a lightweight controller on top of the language backbone to prune low-utility text reasoning tokens. FastV applies a default layer-2 cutoff strategy, discarding redundant visual tokens after the second Transformer layer.

Heima, proposed by Shen et al. (Shen et al., 2025), is a training-required continuous latent-space reasoning baseline. Its core idea is to shift chain-of-thought reasoning into a continuous "hidden thinking" space, compressing each intermediate reasoning step into a single compact thinking token that is decoded only at the final inference step. In the multimodal adaptation used for comparison, the Heima encoder is conditioned on interleaved image and text inputs so that the latent reasoning vector captures fused cross-modal semantics; each intermediate step is encoded as a continuous vector. A frozen Heima decoder reconstructs the final output when needed.

SparseVLM, proposed by Zhang et al. (Zhang et al., 2024), is a training-free visual token sparsification method. Its core mechanism prunes redundant image tokens based on the model's existing attention distribution during the language backbone's forward pass, retaining only the most informative visual patches while discarding background or low-saliency regions. SparseVLM targets only visual tokens and does not address textual token redundancy.

VoT (Visualization-of-Thought) (Wu et al., 2024b) is a purely textual spatial reasoning method. It prompts LLMs to "mentally visualize" spatial states through textual descriptions during reasoning, rather than generating actual image tokens. The core contribution of VoT is demonstrating that explicit spatial visualization thinking can improve LLMs' spatial reasoning capability, though it remains entirely within the text modality.

MVoT (Multimodal Visualization-of-Thought) (Li et al., 2025a) extends the idea of VoT into the truly multimodal domain. Built on Anole-7B, MVoT autoregressively generates discrete visual tokens alongside corresponding textual tokens at each reasoning step, accumulating them in an interleaved fashion into the growing context. This enables the model to perform explicit visual chain-of-thought reasoning beyond the capability of purely text-based approaches.

A.2 Heuristic Compute Optimization Baselines

Gist Token, proposed by Mu et al. (Mu et al., 2023), inserts one or a small number of learnable summary tokens (called "gist tokens") into the input sequence. Through instruction fine-tuning, the model learns to compress and encode the semantic information of the original prompt into the KV representations of these gist tokens. At inference time, only the KV cache entries of the gist tokens are retained while those of the original prompt are discarded, substantially reducing the effective cache length. This method requires training.

Hidden-state norm pruning is a training-free heuristic method that scores each token by the L_2 norm (magnitude) of its hidden-state activation vector at a given layer, removing tokens whose activation norms fall below a threshold—based on the assumption that tokens with small activation magnitudes contribute minimally to the layer's representation. No training or parameter updates are required; the pruning criterion is computed directly from the forward-pass activations.

Heavy-hitter routing, proposed by Zhang et al. (Zhang et al., 2023) as H_2O (Heavy-Hitter Ora-

cle), is a training-free KV cache eviction heuristic. It is based on the observation that a small fraction of tokens consistently accumulates the majority of attention weights across decoding steps (i.e., "heavy-hitter" tokens). H_2O maintains a fixed-size KV cache budget by identifying and retaining tokens with the highest cumulative attention scores, combining the heavy-hitter retention strategy with a recency-based sliding window to prevent recently generated tokens from being fully evicted.

B Dataset Description

MAZE dataset. We construct the MAZE dataset using the official `maze-dataset` (Ivanitskiy et al., 2023) library. Grid mazes of sizes 3–6 are generated with multiple random seeds to ensure layout diversity. For each instance, valid navigation paths are computed and each intermediate step is rendered as a visual thought with directional annotations. Repeated trajectories are removed to prevent train–dev leakage.

MiniBehavior dataset. The MiniBehavior dataset is built upon the official `mini_behavior` (Jin et al., 2023) codebase. A PPO agent is trained in the INSTALLINGAPRINTER environment for room sizes 7, 8, 9, and 10. Successful action sequences are extracted together with rendered intermediate frames, capturing procedural task completion trajectories across varying environment complexities.

FrozenLake dataset. The FrozenLake dataset is constructed using the Visual Spatial Planning environment (Wu et al., 2024a). Q-table–guided trajectories are collected following the success/failure balancing rule described in MVoT, where 50% of trajectories are kept as successful completions and 50% are extended with random actions. Intermediate visual thoughts are rendered in the same format as in MVoT.

C Critical Token Set

As observed in (Wang et al., 2025; Qiu et al., 2025), high-entropy tokens (top 20%) typically correspond to critical decision points across alternative reasoning paths. In contrast, low-entropy tokens (remaining 80%) constitute the more fine-grained and deterministic components of the reasoning process.

Low-entropy tokens are particularly sensitive to KV cache eviction, as maintaining their low-entropy state heavily depends on the availability of preceding contextual information. When critical context is mistakenly evicted, the model tends to produce factual inconsistencies in subsequent low-entropy segments. Such errors can accumulate over long sequences and progressively distort downstream reasoning.

Based on this observation, we define the low-entropy text anchors $\mathcal{K}_t^{\text{text}}$ as the bottom 80% tokens ranked by entropy within text tokens only. This design ensures that deterministically informative tokens, which are essential for maintaining reasoning consistency, are preserved during KV cache compression.

As shown in Figure 2, the definition of spatial anchors $\mathcal{K}_t^{\text{spatial}}$ depends on the underlying multimodal reasoning paradigm. In Volcano-style reasoning, spatial information is explicitly represented via coordinate tokens , which directly encode object locations and spatial relations in a structured form. These tokens naturally serve as spatial anchors, as they provide precise and persistent geometric references across reasoning hops.

In contrast, Anole-style reasoning does not expose explicit coordinate representations. Instead, spatial information is implicitly embedded in visually grounded tokens generated under textual conditioning. We therefore define $\mathcal{K}_t^{\text{spatial}}$ as the top-30% visual tokens in hop t ranked by normalized average attention received from the text queries in the same hop, instantiating the Top-K selection in Equation (9) of the main text with $K = \lfloor 0.3 \times |H_t^{\text{vis}}| \rfloor$.

D Emergent Specialization of Per-Head Eviction Agents

A central claim of ScoutKV is that different KV heads warrant distinct eviction policies. To substantiate this, we visualize the token importance scores produced by the agents on a representative Volcano four-hop spatial reasoning instance (Figure 3) and identify two emergent behavioral patterns that arise entirely from training — neither is encoded in the model architecture or the loss objectives.

Layer-wise modality specialization. Across all examined heads, lower-layer agents assign slightly higher importance to visual tokens than to text tokens, while higher-layer agents shift markedly toward retaining text tokens. This pattern is

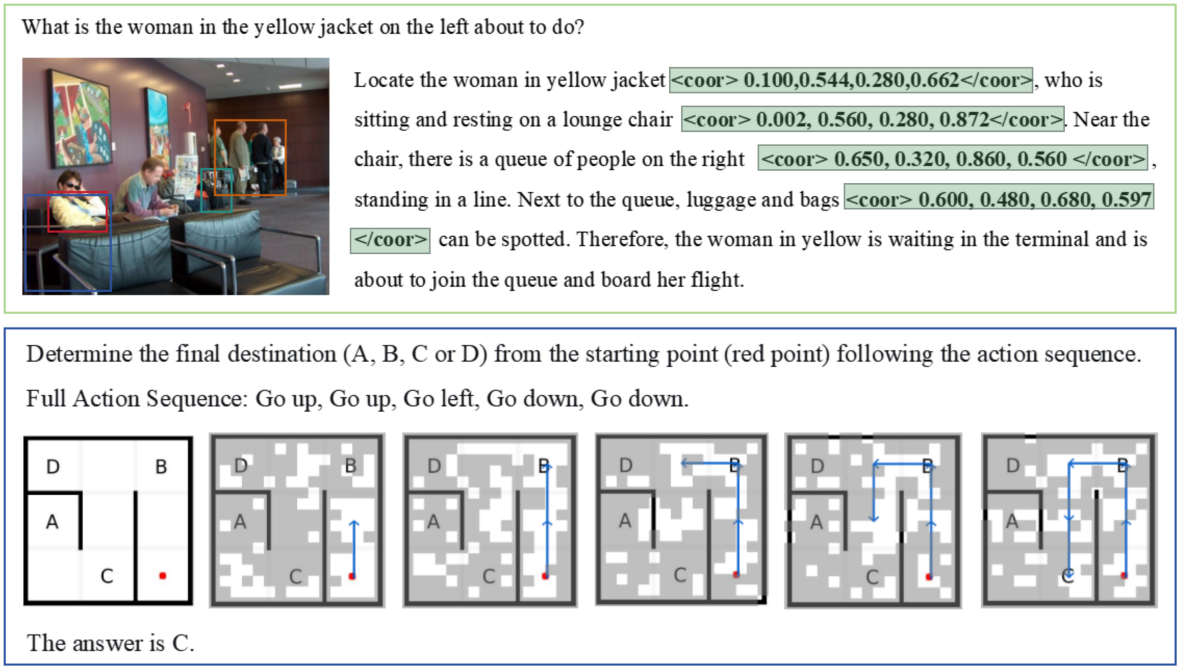

Figure 2: Spatial anchor instantiation across different reasoning paradigms.

consistent with the representational hierarchy of Transformer-based MLLMs: lower layers maintain modality-local features and rely on visual token activations to preserve patch-level spatial detail and bounding-box grounding signals; as depth increases, cross-modal attention progressively distills visual information into the textual stream, after which visual tokens carry diminishing marginal utility for downstream reasoning steps. The modality embedding $e(m_i)$ in Equation (1) supplies the agent with a learnable handle to distinguish visual from textual tokens, but the direction and degree of the resulting preference at each layer emerge from data alone.

Hop-recency bias. Within each panel, tokens originating from more recent reasoning hops consistently receive higher importance scores, visible as darker and taller bars toward the right of each chart. This is intuitively sound: later hops encode spatial conclusions and intermediate deductions that are directly prerequisite to generating the final answer, whereas representations from earlier hops have already been partially integrated into subsequent hidden states. The cross-hop positional encoding $\mathbf{p}_{i,t}$ in Equation (2) gives the agent a continuous signal for hop distance, enabling it to learn this recency preference in a calibrated rather than binary fashion.

Neither pattern was explicitly supervised. The fact that independent per-head agents converge to coherent, interpretable, and structurally distinct eviction strategies — both across layers and across temporal hops — confirms that the behavioral diversity among KV heads is real and consequential. A shared or heuristic policy would flatten this diversity, leading to systematic eviction errors that the per-head design avoids.

E Further Analysis

Table 4: Layer-wise retention behavior in ScoutKV (averaged across reasoning hops).

Layer range	Visual retention	Text retention
0–15	0.54 ~0.81	0.52 ~0.77
16–31	0.28 ~0.62	0.87 ~0.93

This section provides quantitative support for the core analyses presented in the preceding section through three sets of experiments, primarily conducted on the VSR benchmark.

E.1 Layer-wise Modality Retention Behavior (Table 4)

Table 4 quantifies the average visual and textual token retention rates of ScoutKV broken down by layer range on VSR, offering a numerical counter-

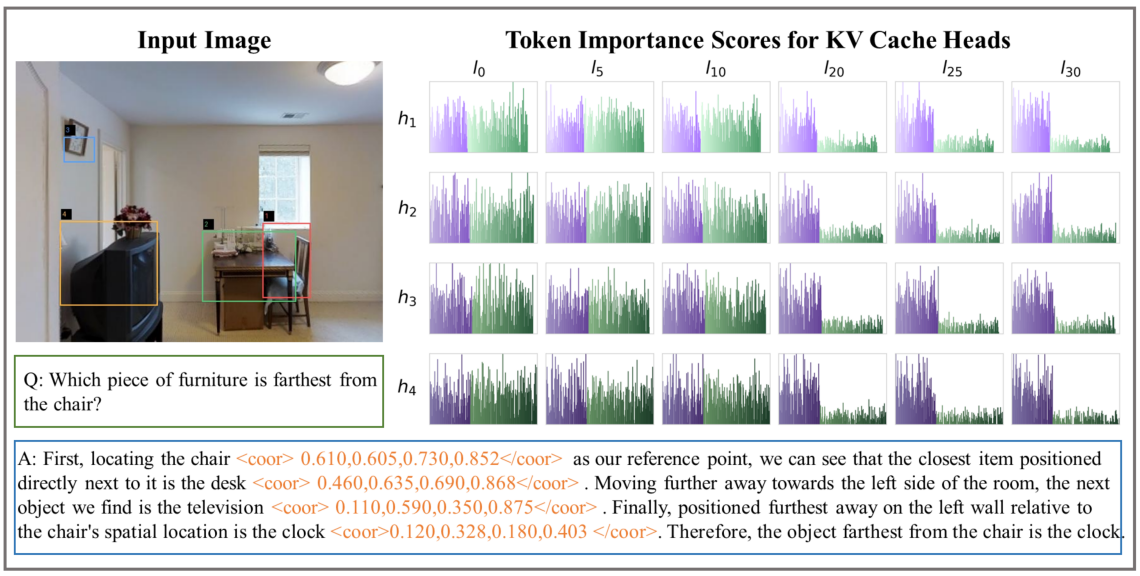

Figure 3: Token importance scores assigned by ScoutKV's per-head scoring agents across six representative layers and four KV heads (h_1–h_4) on a Volcano four-hop spatial reasoning example.

part to the layer-wise modality specialization reported in Figure 3.

In the lower layers (0–15), visual retention ranges from 0.54 to 0.81 and text retention from 0.52 to 0.77—comparable levels, with a slight lean toward visual tokens. This aligns with the well-established representational hierarchy of Transformer-based MLLMs: lower layers maintain modality-local features, and visual tokens at this depth still carry patch-level spatial detail and bounding-box grounding signals that are directly useful for subsequent reasoning. In the upper layers (16–31), the picture shifts markedly: visual retention drops to 0.28–0.62, while text retention rises to 0.87–0.93. This reflects the progressive distillation of visual information into the textual representation stream across deeper layers, after which visual tokens contribute diminishing marginal utility to downstream reasoning steps, while text tokens take on the primary role of maintaining logical coherence.

Notably, these layer-dependent retention preferences are not imposed by the loss objective but emerge from data during training.

E.2 Retention Ratio Ablation (Tables 5 and 6)

Table 5 reports the accuracy–efficiency trade-off on VSR across a range of total token retention ratios. The full-cache baseline achieves FLOPs of 19,842.37 G, latency of 0.63 s, and memory of 8.91 GB.

Table 5: Accuracy–efficiency trade-off under different total token retention ratios. FLOPs are reported relative to the full-token baseline (retention ratio = 1.0). Baseline: FLOPs=19842.37 G, Lat=0.63s, Mem=8.91 GB

Total retention	FLOPs (rel.)	Memory (GB)	Latency (s)	Acc. (%)
0.50	0.56	5.21	0.39	54.2
0.55	0.60	5.65	0.43	60.2
0.60	0.67	6.02	0.46	68.5
0.65	0.72	6.44	0.49	68.1
0.70	0.76	6.81	0.54	68.6

As the retention ratio increases from 0.50 to 0.70, accuracy first rises sharply and then plateaus: 54.2% at 0.50, 60.2% at 0.55, 68.5% at 0.60, and 68.1% and 68.6% at 0.65 and 0.70 respectively—marginal and non-monotonic beyond 0.60. Efficiency, by contrast, degrades consistently with increasing retention: relative FLOPs rise from 0.56 at retention 0.50 to 0.76 at 0.70, with memory and latency scaling similarly. At retention 0.60—the setting adopted in the main experiments—ScoutKV attains 68.5% accuracy while consuming only 0.67× the baseline FLOPs, 6.02 GB memory, and 0.46 s latency, identifying this point as the Pareto-optimal operating condition.

Table 6 disaggregates the total retention budget into the visual and textual components actually allocated by ScoutKV under each setting. At retention 0.60, the learned policy distributes 0.55 to visual tokens and 0.82 to textual tokens—a gap of 0.27 that indicates a consistent preference for preserving text under budget constraints. Across the full range from 0.50 to 0.70, both visual retention

Table 6: Decomposed retention ratios and accuracy under different total token budgets.

Total retention	Visual ret.	Text ret.	Acc. (%)
0.50	0.44	0.77	54.2
0.55	0.50	0.79	60.2
0.60	0.55	0.82	**68.5**
0.65	0.61	0.83	68.1
0.70	0.67	0.85	68.6

($0.44 \rightarrow 0.67$) and text retention ($0.77 \rightarrow 0.85$) increase monotonically, and the asymmetry between the two remains stable across budget levels. This stable asymmetry confirms that the differential treatment of modalities is an intrinsic property learned during training rather than an artifact of any specific budget choice.

E.3 Comparison with Heuristic Baselines (Table 7)

Table 7 compares ScoutKV against three representative heuristic KV cache compression methods on VSR and MAZE, isolating the contribution of policy-driven eviction over hand-crafted scoring rules.

Table 7: Comparison of heuristic methods and ScoutKV on VSR and MAZE benchmarks.

Methods	VSR Acc.	MAZE Acc.	Avg. GFLOPs $\times 10^3$
Gist token	62.8%	84.5%	15.8
Hidden-state norm	57.2%	83.6%	15.3
Heavy-hitter	63.0%	88.7%	14.9
ScoutKV	**68.5%**	**93.7%**	**13.8**

On VSR, ScoutKV achieves 68.5% accuracy, substantially outperforming Heavy-hitter (63.0%), Gist token (62.8%), and Hidden-state norm (57.2%). The accuracy advantage is similarly pronounced on MAZE: ScoutKV reaches 93.7%, compared to 88.7% for Heavy-hitter, 84.5% for Gist token, and 83.6% for Hidden-state norm. In terms of computational cost, ScoutKV records the lowest average GFLOPs of 13.8×10^3 among all methods, undercutting Gist token (15.8×10^3), Hidden-state norm (15.3×10^3), and Heavy-hitter (14.9×10^3) while simultaneously achieving superior accuracy.

These results demonstrate that heuristic scoring rules, which apply a uniform criterion across all KV heads, systematically misallocate the eviction budget in the multi-hop multimodal setting. ScoutKV's per-head policy agents, trained with multi-hop process rewards, overcome this limitation by learning head-specific eviction strategies that are more sensitive to the dynamic informational structure of the reasoning trajectory, yielding consistent gains in both accuracy and efficiency over all heuristic competitors.