WATCH: BUDGETED ACTIVE PERCEPTION AND VISUAL MEMORY CONTROL FOR EFFICIENT MULTIMODAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal efficiency is often treated as a compression problem: process a fixed visual input, then reduce its context. As visual inputs grow in resolution and duration, this view overlooks a fundamental asymmetry: evidence must be acquired before it can be compressed, yet downstream compression cannot recover evidence that was never acquired. Active perception adapts what to observe; token reduction adapts what to retain. Separating these stages misses their causal dependence: the value of new evidence depends on what can survive bounded memory, while memory determines whether further perception is worthwhile. We introduce **WATCH**, a budget-conditioned closed-loop framework for sequential evidence allocation over bounded visual memory. At each step, it halts or acquires a view; only after new evidence arrives does it update bounded persistent memory. WATCH separates cumulative observation cost from memory capacity, enforces both constraints by construction, and learns the structured policy from downstream utility with reinforcement learning; the process admits a budgeted optimal-stopping interpretation and requires a single terminal call to the frozen language backbone. Across image and video benchmarks, WATCH improves the quality–efficiency frontier over passive, compression-only, and active-perception baselines while reducing both visual acquisition and terminal context; gains persist across budgets and backbones and translate into lower latency and memory. The anonymized code is available at `https://anonymous.4open.science/r/WATCH/`.

1 INTRODUCTION

Multimodal large language models (MLLMs) can now reason over increasingly high-resolution images and long videos, but broader visual coverage comes at a substantial computational cost (Chen et al., 2024; Yang et al., 2025a; Liao et al., 2026). A common inference pipeline first processes a predetermined amount of visual input and only then decides how much of the resulting context should reach the language model. This conflates two distinct resources. *Acquiring* visual evidence consumes computation in the visual front-end before a token can ever be reasoned over, whereas *retaining* that evidence determines the visual context that the backbone must ultimately process. Post-hoc token reduction can shrink the latter only after paying for the former; conversely, aggressive low-cost sampling may save acquisition but leave the small region or brief event containing the answer unavailable to downstream reasoning. Recent diagnostics further show that strong MLLMs remain brittle on tasks that require repeated visual observation rather than a single static pass (Zhang et al., 2026). The central efficiency question is therefore not simply how to compress a visual context after it has been seen, but *what is worth observing, what should remain available for reasoning, and when further observation is no longer worth its cost*.

Recent work has begun to address different pieces of this problem. Active visual search and pixel-space reasoning allow models to zoom into informative regions or seek relevant frames rather than relying on a single static visual input (Wu & Xie, 2024; Su et al., 2025; Zheng et al., 2026; Ma et al., 2026a). In parallel, visual-token reduction methods prune or merge redundant representations to reduce the context processed by the language model (Chen et al., 2024; Zhang et al., 2024; Yang et al., 2025a), while newer approaches learn input-side allocation or token-compression policies directly from downstream utility (Liao et al., 2026; Cao et al., 2026; Wang et al., 2026b). These

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.

advances increasingly make both visual acquisition and retention adaptive. Yet they largely optimize different stages of the visual evidence lifecycle in isolation, even though acquisition, retention, and stopping are tightly coupled: the value of a new observation depends on what can survive bounded memory, while the value of retained evidence depends on what has already been observed and what remains worth acquiring.

This coupling points to a more fundamental abstraction: *efficient multimodal reasoning is a sequential evidence-allocation problem*. Perception should continue only when some affordable observation, after passing through the memory bottleneck, is expected to offer more utility than answering now. If it continues, the system must decide where, when, and at what fidelity to observe, then update memory after the new evidence arrives; otherwise, it should halt. Acquisition changes what memory can preserve, memory changes which observations are worth acquiring, and together they determine when perception should end.

We instantiate this view with **WATCH** (**W**hen-to-**A**ttend, **T**oken-**C**ompress, and **H**alt), a budget-conditioned framework for closed-loop multimodal perception. Starting from a fixed low-cost preview, WATCH first decides whether to halt; if not, *Acquire* selects a budget-feasible spatial–temporal view and fidelity, and *Retain* updates bounded memory after the new evidence arrives. Separate budgets B_{obs} and B_{mem} control cumulative visual access and instantaneous visual-memory capacity, respectively. This closed loop admits a budgeted optimal-stopping interpretation, while its memory control admits a task-conditioned rate–distortion interpretation. We train only the lightweight controller with reinforcement learning while keeping the visual front-end and language backbone frozen by default, and invoke the language backbone once on the terminal visual memory.

We evaluate WATCH across diverse image and video reasoning benchmarks. WATCH establishes a stronger aggregate quality–efficiency frontier than passive acquisition, full-input token compression, and specialized active-perception baselines: at the default operating points, it processes only 27–31% of the canonical visual acquisition volume while delivering just 1.3–6.3% of the full visual context to the final backbone call. The gains persist across observation budgets and a second MLLM backbone and translate into substantial latency and memory reductions. Ablations and theory-aligned diagnostics support the complementary roles of Acquire, Retain, and Halt and the predicted budget-dependent stopping and memory-preservation behavior.

Our main contributions are as follows. (1) **A unified formulation of active perception and visual memory control as budgeted evidence allocation.** We formulate efficient multimodal reasoning as closed-loop active perception over a bounded visual memory, explicitly separating cumulative observation cost from persistent memory capacity. The formulation yields a finite-horizon optimal-stopping view and establishes that the optimal halting region expands monotonically as the available observation budget decreases. (2) **A structured Acquire–Retain–Halt policy for joint visual resource control.** WATCH couples budget-feasible visual acquisition with task-conditioned bounded memory and adaptive stopping in a lightweight recurrent controller. Both hard resource constraints are enforced by construction, while policy-invariant dense credit redistributes delayed downstream feedback without changing the optimal policies of the original trajectory objective. (3) **A broad evaluation of the resulting quality–efficiency frontier.** Controlled experiments across image and video reasoning, multiple operating budgets, and two frozen MLLM backbones show that WATCH improves aggregate task quality while reducing both visual acquisition and terminal context. Ablations, theory-aligned diagnostics, and deployment measurements further verify the complementary roles of Acquire, Retain, and Halt and the resulting system-level efficiency gains.

2 Related Work

Active visual perception and pixel-space reasoning. A growing line of work moves multimodal reasoning beyond fixed visual inputs by making what and where to observe part of inference. Guided visual search and visual CoT make task-relevant localization explicit (Wu & Xie, 2024; Shao et al., 2024a), while preference optimization reduces reliance on annotated regions (Zhao et al., 2025). More recent methods learn visual-action trajectories through zooming, cropping, frame selection, multi-step search, query-adaptive video sampling, and iterative evidence seeking (Su et al., 2025; Zhang et al., 2025; Zheng et al., 2026; Zhu et al., 2026; Jeoung et al., 2024; Shen et al., 2025; Carvalho et al., 2026; Ma et al., 2026a; Li et al., 2026; Qiu et al., 2026). Across these variants, the primary control object remains the acquisition or search trajectory. WATCH instead couples acquisition to hard-capacity persistent memory: each new observation competes with stored evidence, and the resulting memory governs both subsequent acquisition and stopping.

Visual token compression and memory control. A complementary literature reduces visual representations after evidence has been encoded. FastV, SparseVLM, and VisionZip exploit attention, language guidance, or redundancy for token reduction (Chen et al., 2024; Zhang et al., 2024; Yang et al., 2025a), while video-oriented methods additionally exploit spatial–temporal redundancy (Huang et al., 2025; Sun et al., 2025; Wang et al., 2025b). Recent methods make preservation richer or sequential by combining task relevance, coverage, diversity, downstream reinforcement learning, query-adaptive selection, or latent memory (Wang et al., 2026c; Cao et al., 2026; Wang et al., 2026a; Ma et al., 2026b; Wang et al., 2026b; Yamao et al., 2026; Yu et al., 2026; Shi et al., 2025). These approaches primarily organize evidence already acquired. In WATCH, *Retain* is instead a recurrent transition inside active perception: each expanded candidate bank is compressed into bounded persistent memory, which in turn changes the value of further acquisition.

Resource-adaptive multimodal inference. Recent methods also move resource control to other stages of inference. ResAdapt, VisionThink, AdaptVision, AutoGaze, Q-Zoom, FlexMem, and VOILA adapt resolution, conditional high-resolution acquisition, pre-backbone visual selection, long-video memory, or pre-retrieval fidelity (Liao et al., 2026; Yang et al., 2025b; Lin et al., 2026; Shi et al., 2026a;b; Chen et al., 2026; Bhope et al., 2026); OmniAgent extends active video control to POMDP-style multimodal interaction with persistent textual memory (Xing et al., 2026). WATCH closes these decisions into a single budget-conditioned feedback loop: it separates cumulative observation cost from hard visual-memory capacity, chooses whether and what to observe from the current compressed evidence, updates memory only after new evidence arrives, and halts when further acquisition is no longer worthwhile. The resulting control object is the *full visual evidence lifecycle under explicit resource constraints*.

3 METHOD: WATCH

Figure 1 summarizes the closed-loop perception process. Given a query q and visual input $\mathcal{V}$, a low-cost, policy-independent preview initializes bounded visual memory, after which a lightweight controller repeatedly coordinates halting, acquisition, and retention. The language backbone π_ϕ is invoked only once, on the terminal memory $\mathbf{M}_K$. See Appendix A for notation and indexing conventions.

3.1 BUDGETED ACTIVE PERCEPTION

Setup and budgets. Let $x = (q, \mathcal{V})$ denote the multimodal input and E_v the visual front-end producing backbone-compatible visual tokens. Before adaptive perception, WATCH applies a fixed, policy-independent low-cost preview $\mathcal{O}_0$. A deterministic reducer $\mathcal{P}_0$ initializes the aligned feature and metadata banks $(\mathbf{M}_0, \mathbf{L}_0) = \mathcal{P}_0(E_v(\mathcal{O}_0(\mathcal{V})); B_{\text{mem}})$, where $\mathbf{M}_0$ contains backbone-compatible visual features and $\mathbf{L}_0$ their aligned controller-side spatial–temporal metadata, with $1 \le |\mathbf{M}_0| = |\mathbf{L}_0| \le B_{\text{mem}}$ for $B_{\text{mem}} \ge 1$. The preview cost c_{pre} is charged to the observation budget, giving $b_0 = B_{\text{obs}} - c_{\text{pre}}$ with $B_{\text{obs}} \ge c_{\text{pre}}$.

WATCH exposes two primary resource controls. The *observation budget* B_{obs} limits cumulative visual access, including the fixed preview, whereas the *memory budget* B_{mem} bounds the visual representations retained for reasoning. A maximum horizon $K_{\max}$ prevents unbounded interaction but is treated as an implementation horizon rather than an additional resource budget.

At step k, the budget-independent evidence state is $z_k = (\mathbf{h}_k, \mathbf{M}_k, \mathbf{L}_k)$, where $\mathbf{h}_k$ recurrently summarizes the query and evidence history, $\mathbf{M}_k$ is the backbone-compatible visual feature bank, and $\mathbf{L}_k$ is its aligned controller-side metadata bank. The metadata are not embedded as additional backbone tokens; WATCH uses them for control and, at the terminal call, deterministically maps their spatial–temporal coordinates to the backbone's native visual position indices (Appendix F.1). Resource information is supplied separately through

$$\boldsymbol{\beta}_k = \left(\frac{b_k}{B_{\text{obs}}^{\max}}, \frac{B_{\text{obs}}}{B_{\text{obs}}^{\max}}, \frac{B_{\text{mem}}}{B_{\text{mem}}^{\max}}, \frac{|\mathbf{M}_k|}{B_{\text{mem}}}, \frac{K_{\max} - k}{K_{\max}} \right), \tag{1}$$

where $B_{\text{obs}}^{\max}$ and $B_{\text{mem}}^{\max}$ are the largest supported budgets. The controller therefore acts on $(z_k, \boldsymbol{\beta}_k)$ without embedding the current budget directly into the evidence state itself.

The control at each step is hierarchical. The controller first chooses $a_k^{\text{halt}} \in \{0, 1\}$. If $a_k^{\text{halt}} = 1$, perception terminates and $y \sim \pi_\phi(\cdot \mid q, \mathbf{M}_k)$. Otherwise, it selects a feasible view v_k; after the resulting visual evidence is encoded, the memory head chooses a_k^{mem} to construct $\mathbf{M}_{k+1}$. This

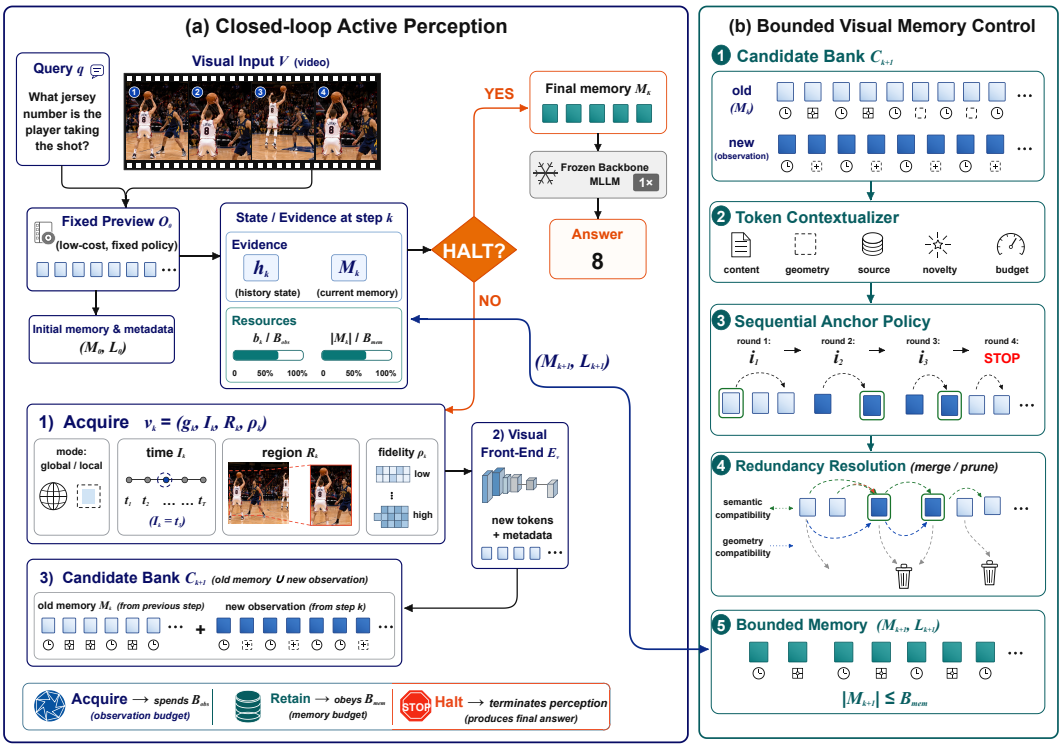


Figure 1: **Overview of WATCH.** The controller iteratively performs Acquire, Retain, and Halt under explicit observation and memory budgets.

ordering reflects the causal structure of active perception: memory control is decided *after* the newly acquired evidence becomes available. The concrete controller is introduced in Section 3.2.

Budgeted objective. For a trajectory τ terminating after $K \le K_{\max}$ adaptive acquisitions, define $C_{\text{obs}}(\tau) = c_{\text{pre}} + \sum_{k=0}^{K-1} c_{\text{obs}}(v_k)$. WATCH optimizes

$$\begin{aligned} &\max_{\theta} \quad \mathbb{E}\left[Q(x,y) - \lambda_{\text{obs}} C_{\text{obs}}(\tau) - \lambda_{\text{mem}} c_{\text{mem}}(\mathbf{M}_K)\right] \\ &\text{s.t.} \quad C_{\text{obs}}(\tau) \le B_{\text{obs}}, \qquad |\mathbf{M}_k| \le B_{\text{mem}} \ \ \forall k, \qquad K \le K_{\max}, \end{aligned} \tag{2}$$

where $Q(x,y)$ measures final task quality and θ contains only the lightweight WATCH parameters. The hard constraints define the admissible deployment envelope, while the nonnegative coefficients λ_{obs} and λ_{mem} favor economical solutions within that envelope. Unless otherwise stated, E_v and the backbone parameters ϕ remain frozen. Our default cost normalization is given in Appendix B.2.

Active perception as budgeted optimal stopping. The central decision is whether another visual observation is worth its cost. For a fixed memory capacity B_{mem}, let $H(z)$ denote the expected utility of answering immediately from evidence state z. Let $\mathcal{F}_{\text{view}}(z,b)$ be the set of views affordable under remaining observation budget b, and let $\widetilde{\mathcal{E}} = (\widetilde{\mathbf{Z}}, \widetilde{\mathbf{L}})$ denote the newly observed visual features together with their aligned controller-side metadata. Let $\Gamma_{B_{\text{mem}}}(z, \widetilde{\mathcal{E}})$ denote all admissible next evidence states obtainable by incorporating observation outcome $\widetilde{\mathcal{E}}$ into a memory of capacity B_{mem}. With ℓ acquisition opportunities remaining,

$$\begin{aligned} V_\ell^\star(z,b) &= \max\left\{ H(z), \sup_{v \in \mathcal{F}_{\text{view}}(z,b)} \left[-\lambda_{\text{obs}} c_{\text{obs}}(v) + \mathbb{E}_{\widetilde{\mathcal{E}}|z,v} \left[\sup_{z' \in \Gamma_{B_{\text{mem}}}(z,\widetilde{\mathcal{E}})} V_{\ell-1}^\star(z', b - c_{\text{obs}}(v)) \right] \right] \right\}, \\ V_0^\star(z,b) &= H(z). \end{aligned} \tag{3}$$

Thus, continuation is optimal only when the expected value of acquiring and subsequently retaining additional evidence exceeds the utility of answering from the current memory.

Proposition 1 (Budget monotonicity and nested halting)**.** *Fix z, B_{mem}, and remaining horizon ℓ. Assume that $\mathcal{F}_{\mathrm{view}}(z, b_1) \subseteq \mathcal{F}_{\mathrm{view}}(z, b_2)$ whenever $b_1 \le b_2$. Then $V_\ell^\star(z, b_1) \le V_\ell^\star(z, b_2)$. Moreover, if halting is optimal at (z, b_2), it is also optimal at (z, b_1). Hence the optimal halting region expands monotonically as the available observation budget decreases.*

Proposition 1 establishes the budget-dependent structure of optimal stopping: reducing the remaining observation budget expands the optimal halting region. This provides a principled foundation for budget-conditioned Halt; Appendix B gives the formulation and proof.

3.2 Acquire–Retain–Halt Controller

Evidence state and adaptive halting. Let $\bar{\mathbf{q}} = E_q(q)$ be a fixed query embedding. A lightweight recurrent controller shared across perception steps updates the evidence summary as

$$\mathbf{h}_k = F_\theta\left(\mathbf{h}_{k-1}, \bar{\mathbf{q}}, \mathrm{Pool}(\mathbf{M}_k, \mathbf{L}_k)\right), \qquad \mathbf{d}_k = [\mathbf{h}_k; E_\beta(\boldsymbol{\beta}_k)], \tag{4}$$

where $\mathrm{Pool}(\mathbf{M}_k, \mathbf{L}_k)$ jointly summarizes the retained visual features and their aligned spatial–temporal metadata, and $\mathbf{d}_k$ is the budget-conditioned decision context used by the policy heads.

The halting head predicts $p_k^{\mathrm{halt}} = \sigma\left(\mathbf{w}_h^\top \mathbf{d}_k + b_h\right)$, $a_k^{\mathrm{halt}} \sim \mathrm{Bernoulli}(p_k^{\mathrm{halt}})$ during training. This head amortizes the stopping comparison in Eq. 3: it should terminate perception when answering from the current evidence is preferable to every affordable continuation.

Acquire: budget-feasible view allocation. Conditional on $a_k^{\mathrm{halt}} = 0$, the view head selects $v_k = (g_k, I_k, R_k, \rho_k) \sim \pi_\theta^{\mathrm{view}}(\cdot \mid \mathbf{d}_k)$, $v_k \in \mathcal{F}_{\mathrm{view}}(z_k, b_k)$, where $g_k \in \{\mathrm{global}, \mathrm{local}\}$ specifies the observation mode, I_k a frame or temporal window, R_k a spatial region, and ρ_k the observation fidelity; for $g_k = \mathrm{global}$, $R_k = R_{\mathrm{full}}$. Discrete masking and continuous reparameterization restrict the policy support to views feasible under the remaining budget.

Executing v_k yields $\widetilde{\mathbf{Z}}_{k+1} = E_v\big(\mathcal{O}(\mathcal{V}, v_k)\big)$, $\widetilde{\mathbf{L}}_{k+1} = \mathrm{Meta}\left(v_k, \widetilde{\mathbf{Z}}_{k+1}\right)$, $b_{k+1} = b_k - c_{\mathrm{obs}}(v_k)$, where Meta maps the selected view and its token layout to normalized spatial–temporal metadata in the coordinate system of the original visual input. The previous and newly acquired evidence form the aligned candidate banks $\mathbf{C}_{k+1} = \left[\mathbf{M}_k; \widetilde{\mathbf{Z}}_{k+1}\right] = \{\mathbf{c}_{k+1,i}\}_{i=1}^{N_{k+1}^c}$, $\mathbf{L}_{k+1}^C = \left[\mathbf{L}_k; \widetilde{\mathbf{L}}_{k+1}\right]$, with $|\mathbf{L}_{k+1}^C| = N_{k+1}^c$. Hence previously retained and newly acquired evidence compete under the same capacity constraint rather than receiving privileged treatment based on their age or acquisition cost.

Retain: novelty- and coverage-aware memory control. The aligned metadata bank $\mathbf{L}_{k+1}^C = \{\boldsymbol{\ell}_{k+1,i}\}_{i=1}^{N_{k+1}^c}$ provides the spatial–temporal metadata of each candidate, while $\chi_{k+1,i} \in \{\mathrm{old}, \mathrm{new}\}$ records its provenance. Let $\bar{\mathbf{c}}_{k+1,i} = P_c(\mathbf{c}_{k+1,i})$ be a lightweight controller projection. For a newly acquired token, we measure novelty relative to the current memory by

$$\nu_{k+1,i} = \frac{1}{2}\left[1 - \max_{\mathbf{m} \in \mathbf{M}_k} \cos\left(\bar{\mathbf{c}}_{k+1,i}, P_c(\mathbf{m})\right)\right], \qquad \mathbf{c}_{k+1,i} \in \widetilde{\mathbf{Z}}_{k+1}, \tag{5}$$

with $\nu_{k+1,i} = 1$ if $\mathbf{M}_k = \varnothing$ and $\nu_{k+1,i} = 0$ for inherited tokens. Unlike a frame-to-frame residual, this memory-relative signal requires no token-wise alignment and therefore applies to irregular crops, repeated zooms, and temporally nonadjacent observations.

After acquisition, let $\mathbf{d}_k^+$ denote the decision context obtained from $\mathbf{h}_k$ using the updated remaining budget b_{k+1} and remaining horizon. A shallow token contextualizer jointly processes candidate content, geometry, provenance, novelty, and this post-acquisition resource context:

$$\mathbf{r}_{k+1,i} = [P_c(\mathbf{c}_{k+1,i}); E_\ell(\boldsymbol{\ell}_{k+1,i}); E_\chi(\chi_{k+1,i}); \nu_{k+1,i}], \quad \{\mathbf{u}_{k+1,i}\}_{i=1}^{N_{k+1}^c} = G_\theta\left(\{\mathbf{r}_{k+1,i}\}_{i=1}^{N_{k+1}^c}, \mathbf{d}_k^+\right). \tag{6}$$

The memory head uses a capacity-constrained autoregressive anchor policy whose support directly satisfies the memory limit. Starting from an empty anchor set, it repeatedly selects an unselected candidate or a STOP action. STOP is disabled for the first selection, and the procedure terminates no later than B_{mem} anchors. Hence the memory capacity is satisfied by policy support itself: $a_k^{\mathrm{mem}} \sim \pi_\theta^{\mathrm{mem}}\left(\cdot \mid \{\mathbf{u}_{k+1,i}\}_{i=1}^{N_{k+1}^c}, \mathbf{d}_k^+\right)$, $1 \le |\mathcal{S}_{k+1}| \le B_{\mathrm{mem}}$, where $\mathcal{S}_{k+1}$ is the resulting anchor set.

Non-anchor tokens are merged only when they are both semantically redundant with and geometrically compatible with a retained anchor; otherwise they are pruned. Thus each candidate has one of

three effective outcomes—retain, merge, or prune—while reinforcement learning operates over the substantially smaller anchor-selection space. The resulting aligned banks satisfy

$$(\mathbf{M}_{k+1}, \mathbf{L}_{k+1}) = \mathcal{K}\left(\mathbf{C}_{k+1}, \mathbf{L}^{C}_{k+1}, a^{\mathrm{mem}}_{k}\right), \qquad 1 \le |\mathbf{M}_{k+1}| = |\mathbf{L}_{k+1}| \le B_{\mathrm{mem}}. \tag{7}$$

The compressed feature bank and its aligned metadata then produce the next evidence state through Eq. 4, closing the Acquire–Retain–Halt loop. From another perspective, the memory update admits a task-conditioned rate–distortion interpretation: it seeks a compact visual context that preserves the answer-relevant distribution induced by the candidate evidence. Appendix C gives the full budget-feasible view parameterization, sequential anchor policy, geometry-aware redundancy resolution, and rate–distortion interpretation.

3.3 Learning and Inference

WATCH is naturally trained with reinforcement learning because halting and memory control are discrete, view allocation contains non-differentiable visual operations, and the utility of intermediate decisions is revealed only through delayed downstream task feedback.

Trajectory reward. For each training example and prescribed budget pair $(B_{\mathrm{obs}}, B_{\mathrm{mem}})$, we sample G complete WATCH trajectories from the behavior policy. Rollout g receives the terminal utility

$$R_g = Q(x, y^{(g)}) - \lambda_{\mathrm{obs}} C_{\mathrm{obs}}(\tau^{(g)}) - \lambda_{\mathrm{mem}} c_{\mathrm{mem}}\left(\mathbf{M}^{(g)}_{K_g}\right), \tag{8}$$

which is exactly the unconstrained component of Eq. 2. Rollouts within a group share the same input, fixed preview, and resource budget, so their relative returns directly compare alternative Acquire–Retain–Halt strategies under the same operating condition.

Potential-based dense credit. A terminal utility alone gives weak credit to a sequence of acquisition and memory decisions. We therefore introduce a lightweight training-time potential Φ_ξ defined on *policy-parameter-independent shaping states*. Let $\bar{\zeta}^{M}_{k} = (\mathrm{M}, q, \mathbf{M}_k, \mathbf{L}_k, \boldsymbol{\beta}_k)$ and $\bar{\zeta}^{C}_{k+1} = (\mathrm{C}, q, \mathbf{C}_{k+1}, \mathbf{L}^{C}_{k+1}, \boldsymbol{\beta}^{+}_{k})$ denote the pre-acquisition and post-acquisition shaping states, respectively. The potential parameters ξ are disjoint from the WATCH policy parameters θ and are held fixed during each controller update.

The shaping terms for the two internal transitions are

$$F^{\mathrm{acq}}_{k} = \Phi_\xi(\bar{\zeta}^{C}_{k+1}) - \Phi_\xi(\bar{\zeta}^{M}_{k}), \quad F^{\mathrm{mem}}_{k} = \Phi_\xi(\bar{\zeta}^{M}_{k+1}) - \Phi_\xi(\bar{\zeta}^{C}_{k+1}), \tag{9}$$

with terminal shaping $F^{\mathrm{halt}} = -\Phi_\xi(\bar{\zeta}^{M}_{K})$.

Proposition 2 (Policy-invariant dense credit)**.** *For any fixed potential Φ_ξ and scalar $\eta \ge 0$, augmenting the trajectory reward with η times the shaping terms in Eq. 9 and F^{halt} preserves the set of optimal WATCH policies for a fixed input and prescribed budget, since*

$$\sum_{k=0}^{K-1} (F^{\mathrm{acq}}_{k} + F^{\mathrm{mem}}_{k}) + F^{\mathrm{halt}} = -\Phi_\xi(\bar{\zeta}^{M}_{0}), \tag{10}$$

which is independent of the adaptive actions and the policy being compared.

Thus, Φ_ξ redistributes delayed feedback across acquisition and memory control without altering the underlying trajectory objective. We train it to approximate *continuation utility*, treating costs incurred before a shaping state as sunk. Appendix D gives the complete estimator and proof.

Group-relative policy update. Following the same-context group normalization used in GRPO (Shao et al., 2024b), we construct stage-specific relative advantages for WATCH's heterogeneous decisions. For controller stage $d \in \{\mathrm{halt}, \mathrm{view}, \mathrm{mem}\}$, let $G^{d}_{g,k}$ denote the realized continuation return and $\bar{\zeta}^{d}_{g,k}$ its shaping state. We form $\widetilde{G}^{d}_{g,k} = G^{d}_{g,k} - \eta\Phi_\xi(\bar{\zeta}^{d}_{g,k})$ and normalize it across group rollouts reaching the same decision stage to obtain $A^{d}_{g,k}$.

For each stochastic policy factor u—a halt decision, a view decision, or one autoregressive memory decision—with context $\psi_{g,u}$, define $\rho_{g,u}(\theta) = \frac{\pi_\theta(a_{g,u}|\psi_{g,u})}{\pi_{\theta_{\mathrm{old}}}(a_{g,u}|\psi_{g,u})}$. WATCH minimizes the clipped surrogate

$$\mathcal{L}_{\mathrm{RL}} = -\mathbb{E}_{g,u}\left[\min\left(\rho_{g,u}A_{g,u}, \mathrm{clip}(\rho_{g,u}, 1-\varepsilon, 1+\varepsilon)A_{g,u}\right)\right], \tag{11}$$

where $A_{g,u}$ is the controller-stage advantage associated with factor u (Schulman et al., 2017). Action-specific likelihood ratios, autoregressive memory-sequence handling, and the complete regularized objective are given in Appendix D.3.

We warm-start the controller from budget-conditioned offline demonstrations before on-policy optimization. Demonstration construction, continuation-value learning, and the full training schedule are given in Appendix D. The visual front-end and language backbone remain frozen by default.

Deterministic inference. At deployment, WATCH determinizes all controller decisions over the same budget-feasible supports; τ_{halt} is validation-selected. Algorithm 1 summarizes the resulting Acquire–Retain–Halt procedure, after which the language backbone is invoked once on the terminal memory. Detailed deterministic decoding, resource accounting, and complexity analysis are provided in Appendix E.

Algorithm 1 WATCH: Budgeted Acquire–Retain–Halt Inference

Require: query q, visual input $\mathcal{V}$, budgets $(B_{\mathrm{obs}}, B_{\mathrm{mem}})$, horizon $K_{\max}$
$(\mathbf{M}_0, \mathbf{L}_0)$, $b_0 \leftarrow$ INITIALIZE$(\mathcal{V}, B_{\mathrm{obs}}, B_{\mathrm{mem}})$; initialize $\mathbf{h}_0, \mathbf{d}_0$; $k \leftarrow 0$
while $k < K_{\max}$ and $\mathcal{F}_{\mathrm{view}}(z_k, b_k) \neq \varnothing$ **do**
  if $p_k^{\mathrm{halt}} \geq \tau_{\mathrm{halt}}$ **then**
    break ▷ *Halt*
  end if
  $v_k \leftarrow$ DETERMINISTICVIEW$_\theta(\mathbf{d}_k, b_k)$ ▷ *Acquire*
  $(\widetilde{\mathbf{Z}}_{k+1}, \widetilde{\mathbf{L}}_{k+1}, b_{k+1}) \leftarrow$ ACQUIRE$(\mathcal{V}, v_k, b_k)$
  $(\mathbf{C}_{k+1}, \mathbf{L}^C_{k+1}) \leftarrow ([\mathbf{M}_k; \widetilde{\mathbf{Z}}_{k+1}], [\mathbf{L}_k; \widetilde{\mathbf{L}}_{k+1}])$
  Form post-acquisition context $\mathbf{d}_k^+$
  $(\mathbf{M}_{k+1}, \mathbf{L}_{k+1}) \leftarrow$ MEMORYCONTROL$_\theta(\mathbf{C}_{k+1}, \mathbf{L}^C_{k+1}, \mathbf{d}_k^+, B_{\mathrm{mem}})$ ▷ *Retain*
  Update $\mathbf{h}_{k+1}, \mathbf{d}_{k+1}$; $k \leftarrow k+1$
end while
return $y \sim \pi_\phi(\cdot \mid q, \mathbf{M}_k)$

4 EXPERIMENTS

Evaluation scope. We evaluate WATCH as a joint visual resource-control policy rather than a terminal token compressor in isolation. Our experiments examine whether WATCH improves the quality–efficiency frontier across image and video reasoning, whether Acquire, Retain, and Halt provide complementary benefits, whether the learned policy exhibits behavior consistent with the budgeted active-perception formulation, and whether the resulting resource savings translate into system-level efficiency. Throughout, we distinguish the cost of *acquiring* visual evidence from the cost of *retaining* the final visual context, since reducing one does not in general reduce the other.

4.1 EXPERIMENTAL SETUP

Backbone and benchmarks. We use Qwen2.5-VL-7B-Instruct (Bai et al., 2025b) and Qwen3-VL-8B-Instruct (Bai et al., 2025a) as the primary frozen backbone, allowing the experiments to isolate the effect of the WATCH perception policy. Image evaluation covers fine-grained visual search with V*Bench (Wu & Xie, 2024), high-resolution perception with HR-Bench-4K/8K (Wang et al., 2025a), document understanding with DocVQA (Mathew et al., 2021), and general multimodal reasoning with RealWorldQA (xAI, 2024) and MMMU (Yue et al., 2024). Video evaluation uses Video-MME (Fu et al., 2025), LongVideoBench (Wu et al., 2024), EgoSchema (Mangalam et al., 2023), and NExT-QA (Xiao et al., 2021).

Baselines. We compare against four complementary families. Passive baselines include Full Input, Preview Only, and matched-cost Uniform Budget. Visual-token reduction baselines include FastV (Chen et al., 2024), SparseVLM (Zhang et al., 2024), VisionZip (Yang et al., 2025a), TPRL (Cao et al., 2026), and SCORE (Wang et al., 2026b). Active image-perception baselines include V* (Wu & Xie, 2024) and ZoomEye (Shen et al., 2025), with DeepEyes (Zheng et al., 2026) and AdaptVision (Lin et al., 2026) included as released active-model references; for video, we additionally compare with Video Active Perception (VAP) (Ma et al., 2026a) and VideoTree (Wang et al., 2025c). Whenever the method admits a faithful port, we use the same frozen backbone, visual interface, and decoding protocol; otherwise, published-model results are reported separately from controlled comparisons.

Budgets and evaluation. Unless otherwise specified, the default operating point is $(B_{\mathrm{obs}}, B_{\mathrm{mem}}) = (2, 128)$ for images and $(16, 256)$ for videos. We report task quality together with cumulative observation cost and the final visual context delivered to the backbone, and complement these proxy resources with measured latency and memory. Seed variability is reported in

Appendix G.3. Complete data, training, budget, baseline, decoding, and system-measurement protocols are provided in Appendix F; full per-benchmark results and the Qwen3-VL-8B transfer study are reported in Appendix G.

4.2 Quality–Efficiency Trade-offs

Default operating point. Table 1 compares task quality with visual acquisition and terminal context. On images, WATCH attains the strongest controlled macro-average (77.7) with $r_{\mathrm{acq}} = 0.31$ and $r_{\mathrm{ctx}} = 0.063$. Compression-only methods can reduce the terminal context but still incur full acquisition, while specialized search remains stronger on V*Bench. AdaptVision provides a recent active-acquisition reference at a similar acquisition scale, but its MLLM is itself post-trained for adaptive perception and is therefore excluded from the frozen-backbone ranking. Thus WATCH's advantage is aggregate rather than task-specific.

Table 1: **Main quality–efficiency comparison with Qwen2.5-VL-7B-Instruct.** Gen. is the mean of RealWorldQA and MMMU, whereas Avg. is the macro-average over all six image evaluation settings (or all four video benchmarks). r_{acq} measures the canonical-equivalent amount of visual evidence processed by all acquisition or selector modules, including selector-side scans, whereas r_{ctx} measures the terminal visual context delivered to the answering backbone. For methods using the common Qwen visual front-end, r_{acq} reduces to $N_{\mathrm{enc}}/N_{\mathrm{enc}}^{\mathrm{full}}$. Bold and underlined entries denote the best and second-best results among controlled comparisons, respectively. † denotes a released-model reference whose MLLM itself is trained for active perception and is therefore excluded from matched-frozen-backbone ranking.

Image reasoning WATCH default: $(B_{\mathrm{obs}}, B_{\mathrm{mem}}) = (2, 128)$

Method	V*	HR-4K	HR-8K	DocVQA	Gen.	Avg.	$r_{\mathrm{acq}}\downarrow$	$r_{\mathrm{ctx}}\downarrow$
Full Input	71.2	68.8	65.3	**93.1**	<u>63.3</u>	70.8	1.00	1.000
Uniform Budget	75.4	70.9	67.8	90.9	62.1	71.5	**0.31**	<u>0.100</u>
SparseVLM	71.9	69.2	65.9	92.0	62.7	70.7	1.00	0.223
VisionZip	72.6	69.7	66.4	92.6	62.9	71.2	1.00	<u>0.100</u>
TPRL	73.0	70.2	67.1	<u>92.8</u>	63.1	71.5	1.00	<u>0.100</u>
V*	**91.0**	74.8	71.4	91.7	62.5	75.7	0.74	–
ZoomEye	<u>90.7</u>	<u>76.1</u>	<u>73.2</u>	92.2	63.0	<u>76.4</u>	<u>0.66</u>	–
DeepEyes†	90.1	75.1	72.6	93.5	64.8	76.8	0.58	–
AdaptVision†	89.4	76.8	73.9	92.6	63.5	76.6	0.35	–
WATCH	90.2	**78.2**	**75.8**	**93.1**	**64.5**	**77.7**	**0.31**	**0.063**

Video reasoning WATCH default: $(B_{\mathrm{obs}}, B_{\mathrm{mem}}) = (16, 256)$

Method	Video-MME	LongVideoBench	EgoSchema	NExT-QA	Avg.	$r_{\mathrm{acq}}\downarrow$	$r_{\mathrm{ctx}}\downarrow$
Full Input	65.8	55.4	60.8	78.9	65.2	<u>1.00</u>	1.000
Uniform Budget	62.2	50.6	56.9	76.2	61.5	**0.27**	<u>0.016</u>
SparseVLM	64.7	53.9	60.3	78.0	64.2	<u>1.00</u>	0.125
VisionZip	65.0	54.7	60.0	78.4	64.5	<u>1.00</u>	<u>0.016</u>
SCORE	<u>66.4</u>	<u>56.7</u>	**63.4**	79.8	<u>66.6</u>	<u>1.00</u>	<u>0.016</u>
VideoTree	63.2	54.4	61.1	75.6	63.6	<u>1.00</u>	0.188
VAP	65.6	56.0	<u>63.3</u>	<u>80.1</u>	66.3	<u>1.00</u>	0.125
WATCH	**67.5**	**58.0**	62.9	**81.2**	**67.4**	**0.27**	**0.013**

The video comparison gives WATCH the strongest controlled macro-average (67.4) with $r_{\mathrm{acq}} = 0.27$ and $r_{\mathrm{ctx}} = 0.013$. SCORE has the highest EgoSchema score and uses full acquisition. Complete per-benchmark comparisons, including Preview Only, FastV, the unconstrained high-resolution reference, second-backbone results, and additional recent adaptive-acquisition references are reported in Appendix G.

Across observation budgets. Figure 2 shows that WATCH consistently outperforms matched-cost uniform acquisition across the evaluated budgets and exhibits diminishing returns as additional budget exposes less new answer-relevant evidence. Relative to active baselines, WATCH reaches a comparable or better quality range with substantially less visual acquisition.

These proxy savings also translate into measured system efficiency: at the default operating point on a single H100, WATCH reduces E2E latency from 3.22 s to 1.63 s on images and from 10.51 s to 4.11 s on videos relative to Full Input, while adding only 0.09 s and 0.20 s, respectively, over matched-cost Uniform Budget. Complete budget sweeps and deployment measurements are reported in Appendix H.

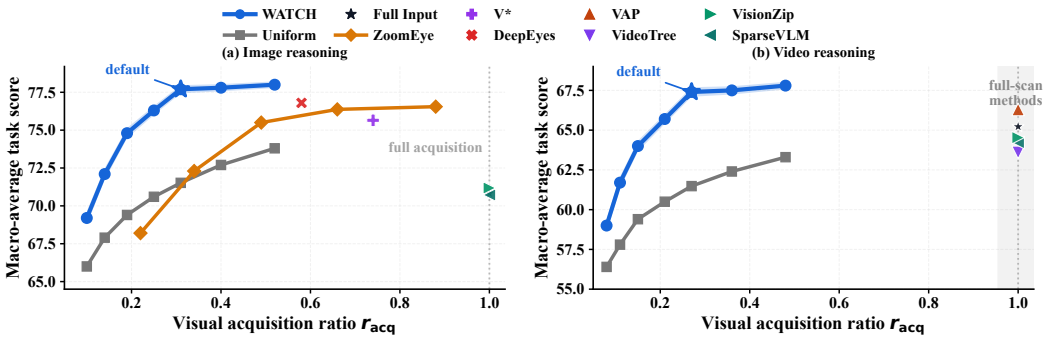


Figure 2: **Quality–acquisition trade-offs.** For WATCH and Uniform Budget, connected markers vary B_{obs} with $B_{\text{mem}} = 128$ for images and 256 for videos; active-search markers show measured operating points under their native sensing budgets. r_{acq} counts all visual evidence processed by acquisition or selector modules, so full-scan selectors such as VAP and VideoTree remain at $r_{\text{acq}} = 1$ even when only a subset of frames reaches the answering model. Shaded regions show variation across three WATCH controller seeds; lines connect measured operating points.

4.3 Analysis of WATCH

Where do the gains come from? Table 2 shows that Acquire, Retain, and Halt play complementary roles. Replacing adaptive acquisition with uniform views causes the largest quality degradation despite slightly greater visual acquisition, identifying active evidence acquisition as the primary quality driver. Static Retain also degrades performance while using a larger terminal context, showing that *which* evidence survives matters beyond memory capacity. In contrast, disabling halting changes quality little but substantially increases acquisition. Halt therefore mainly improves the operating point by stopping once further perception is no longer worth its expected cost.

Table 2: **Decomposing Acquire, Retain, and Halt.** All variants use WATCH's hard budgets. *Uniform Acquire* follows a fixed query-independent coverage schedule, *Static Retain* uses deterministic geometry-preserving reduction, and *No Halt* continues whenever a view is affordable. ΔQ is the macro-average change from WATCH; resource columns report the corresponding resource values.

	Image				Video			
Variant	Avg.↑	ΔQ	$r_{\text{acq}}\downarrow$	$r_{\text{ctx}}\downarrow$	Avg.↑	ΔQ	$r_{\text{acq}}\downarrow$	$r_{\text{ctx}}\downarrow$
Uniform Acquire	72.8	−4.9	0.34	0.066	62.4	−5.0	0.31	0.014
Static Retain	75.2	−2.5	0.33	0.100	64.1	−3.3	0.30	0.016
No Halt	**77.8**	+0.1	0.44	0.067	**67.6**	+0.2	0.38	0.014
WATCH	77.7	–	**0.31**	**0.063**	67.4	–	**0.27**	**0.013**

Does the learned policy match the formulation? The more detailed diagnostics are consistent with the intended structure of WATCH. Counterfactual budget tests show approximately nested stopping decisions as the available observation budget decreases; continuation states concentrate above zero in empirical continuation advantage, whereas halted states concentrate below zero; and the learned memory policy achieves lower teacher-forced query-conditioned answer-distribution distortion under the same capacity. Complete ablations are provided in Appendix I, theory-aligned analyses in Appendix J, and trajectory visualizations and failure cases in Appendix K.

5 Conclusion

Our results support efficient multimodal reasoning as *sequential evidence allocation* rather than post-hoc token reduction: by coupling selective acquisition, bounded memory, and adaptive stopping under explicit observation and memory budgets, WATCH improves the quality–efficiency frontier across image and video reasoning. As visual inputs grow in resolution and duration, scalable multimodal reasoning may increasingly depend on the ability to *observe selectively, remember selectively, and stop adaptively*. Our current implementation assumes a predefined observation interface and backbone-compatible visual memory; extending WATCH to richer sensing and action spaces and more transferable cross-backbone controllers is a natural next step.

AI USE STATEMENT

We used OpenAI's ChatGPT as a research-support tool for early-stage brainstorming and feedback on high-level research framing and methodological presentation, literature discovery and bibliographic checking, and drafting, revising, and polishing limited portions of the manuscript. It was also used for final consistency and minor-error checks. Generative AI was not used to generate source code, run experiments, or produce experimental data or numerical results. All scientific decisions, experimental protocols, analyses, and conclusions were made by the authors. All cited sources were selected by the authors and independently verified against primary publisher or project records, and all AI-assisted text and suggestions were reviewed, verified, and revised by the authors. The authors take full responsibility for the technical content, claims, citations, and final presentation of this work.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide an **anonymized code repository** at `https://anonymous.4open.science/r/WATCH/`, and additionally include a snapshot of the anonymized code as part of the **supplementary material**. The code contains the main implementation of WATCH together with detailed instructions for setup, training, inference, and evaluation. It covers the principal components of the framework, including budget-feasible visual acquisition, visual memory selection and merging, the Qwen2.5-VL adapter, offline demonstration construction, behavior cloning (BC), potential-based reinforcement learning (RL), and the corresponding evaluation pipeline.

The manuscript further specifies the method, implementation, and evaluation protocols in detail. Section 3 and Appendices A–E provide the notation, complete formulation and proofs, controller parameterization, learning objectives and training procedure, deterministic inference protocol, and complexity accounting. Appendix F documents the backbones, benchmark protocols, controller-training data, observation operators and budgets, model hyperparameters, training schedule, baseline implementations, system measurements, and statistical procedures. Appendices G–K provide full benchmark results, seed variability and confidence intervals, budget and system-efficiency measurements, ablations and training diagnostics, theory-aligned analyses, and qualitative trajectories and failure cases. Together, the anonymized code release and the detailed experimental documentation are intended to support independent reproduction of the reported results.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report, 2025a. URL `https://arxiv.org/abs/2511.21631`.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025b. URL `https://arxiv.org/abs/2502.13923`.

Rahul Atul Bhope, K. R. Jayaram, Vinod Muthusamy, Ritesh Kumar, Vatche Isahagian, and Nalini Venkatasubramanian. VOILA: Value-of-information guided fidelity selection for cost-aware multimodal question answering, 2026. URL `https://arxiv.org/abs/2602.03007`.

Sihan Cao, Jianwei Zhang, Pengcheng Zheng, Jiaxin Yan, Caiyan Qin, Yalan Ye, Wei Dong, Peng Wang, Yang Yang, and Chaoning Zhang. Language-guided token compression with reinforcement learning in large vision-language models. *arXiv preprint arXiv:2603.13394*, 2026.

Miguel Carvalho, Hélder Dias, and Bruno Martins. CropVLM: Learning to zoom for fine-grained vision-language perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 11170–11179, June 2026.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 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*, pp. 19–35. Springer, 2024.

Tao Chen, Kun Zhang, Qiong Wu, Xiao Chen, Chao Chang, Xiaoshuai Sun, Yiyi Zhou, and Rongrong Ji. Scaling the long video understanding of multimodal large language models via visual memory mechanism. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 31877–31888, June 2026.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, volume 2024, pp. 35549–35562, 2024.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

Xiaohu Huang, Hao Zhou, and Kai Han. PruneVid: Visual token pruning for efficient video large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19959–19973, 2025.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. TGIF-QA: Toward spatio-temporal reasoning in visual question answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1359–1367. IEEE, 2017.

Sullam Jeoung, Goeric Huybrechts, Bhavana Ganesh, Aram Galstyan, and Sravan Bodapati. Adaptive video understanding agent: Enhancing efficiency with dynamic frame sampling and feedback-driven reasoning. *arXiv preprint arXiv:2410.20252*, 2024.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pp. 235–251. Springer, 2016.

Keliang Li, Yansong Li, Hongze Shen, Mengdi Liu, Hong Chang, and Shiguang Shan. LensWalk: Agentic video understanding by planning how you see in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19518–19528, June 2026.

Huanxuan Liao, Zhongtao Jiang, Yupu Hao, Yuqiao Tan, Shizhu He, Ben Wang, Jun Zhao, Kun Xu, and Kang Liu. ResAdapt: Adaptive resolution for efficient multimodal reasoning. *arXiv preprint arXiv:2603.28610*, 2026.

Zichuan Lin, Yicheng Liu, Yang Yang, Lvfang Tao, and Deheng Ye. AdaptVision: Efficient vision-language models via adaptive visual acquisition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11923–11932, June 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

Martin Q Ma, Willis Guo, Aditya Agrawal, Ankit Gupta, Paul Pu Liang, Ruslan Salakhutdinov, and Louis-Philippe Morency. Video active perception: Effective inference-time long-form video understanding with vision-language models. *arXiv preprint arXiv:2605.01662*, 2026a.

Yinchao Ma, Qiang Zhou, Zhibin Wang, Xianing Chen, Hanqing Yang, Jun Song, and Bo Zheng. Contribution-aware token compression for efficient video understanding via reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24317–24325, 2026b.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pp. 2263–2279, 2022.

Minesh Mathew, Dimosthenis Karatzas, and C.V. Jawahar. DocVQA: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2200–2209, January 2021.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In Ivan Bratko and Saso Dzeroski (eds.), *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, pp. 278–287. Morgan Kaufmann, 1999.

Jihao Qiu, Lingxi Xie, Xinyue Huo, Qi Tian, and Qixiang Ye. LongVideo-R1: Smart navigation for low-cost long video understanding, 2026. URL https://arxiv.org/abs/2602.20913.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual CoT: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024a.

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

Haozhan Shen, Kangjia Zhao, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, Mingwei Zhu, and Jianwei Yin. ZoomEye: Enhancing multimodal llms with human-like zooming capabilities through tree-based image exploration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6602–6618, 2025.

Baifeng Shi, Stephanie Fu, Long Lian, Hanrong Ye, David Eigen, Aaron Reite, Jan Kautz, Boyi Li, David M Chan, Trevor Darrell, et al. Attend before attention: Efficient and scalable video understanding via autoregressive gazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17022–17034, 2026a.

Yuheng Shi, Xiaohuan Pei, Linfeng Wen, Minjing Dong, and Chang Xu. Q-Zoom: Query-aware adaptive perception for efficient multimodal large language models, 2026b. URL https://arxiv.org/abs/2604.06912.

Yumeng Shi, Quanyu Long, and Wenya Wang. Static or dynamic: Towards query-adaptive token selection for video question answering. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10759–10771, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.545. URL https://aclanthology.org/2025.emnlp-main.545/.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8317–8326. IEEE, 2019.

Jiafei Song, Fengwei Zhou, Jin Qu, Wenjin Jason Li, Tong Wu, Gengjian Xue, Zhikang Zhao, Daomin Wei, Yichao Lu, and Bailin Na. EvoComp: Learning visual token compression for multimodal large language models via semantic-guided evolutionary labeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3532–3542, June 2026.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel Reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-0280. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/0c38f54740062529aa4117a04b583f3c-Paper-Conference.pdf.

Boyuan Sun, Jiaxing Zhao, Xihan Wei, and Qibin Hou. LLaVA-Scissor: Token compression with semantic connected components for video llms. *arXiv preprint arXiv:2506.21862*, 2025.

Xiaopeng Tan, Qingzhe Pan, Kai Zhang, and Xuemei Xie. Human-inspired agentic video understanding via coarse-to-fine spatiotemporal exploration. *Neurocomputing*, pp. 134985, 2026. ISSN 0925-2312. doi: 10.1016/j.neucom.2026.134985. URL https://www.sciencedirect.com/science/article/pii/S0925231226023830.

Hengyi Wang, Weiying Xie, Hui Jiang, Yaotao Wei, Kai Jiang, Mingxiang Cao, Chenhe Hao, and Leyuan Fang. TOP-RL: Task-optimized progressive token pruning with reinforcement learning for vision language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15824–15832, 2026a.

Shida Wang, YongXiang Hua, Zhou Tao, Haoyu Cao, and Linli Xu. Dynamic token compression for efficient video understanding through reinforcement learning. *arXiv preprint arXiv:2603.26365*, 2026b.

Tinghao Wang, Yichen Guo, Rui Huang, Zheng Lu, Qizhe Zhang, Chenxi Li, Yuan Zhang, Jiajun Cao, Zhirong Shen, Yaosong Du, et al. TOPS: First-principles visual token pruning via constructing token optimal preservation sets for efficient mllm inference. *arXiv preprint arXiv:2606.27161*, 2026c.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7907–7915, 2025a.

Xiao Wang, Qingyi Si, Shiyu Zhu, Jianlong Wu, Li Cao, and Liqiang Nie. AdaReTaKe: Adaptive redundancy reduction to perceive longer for video-language understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5417–5432, 2025b.

Yajing Wang, Chao Bi, Junshu Sun, Shufan Shen, Zhaobo Qi, Shuhui Wang, and Qingming Huang. ActiveScope: Actively seeking and correcting perception for mllms, 2026d. URL https://arxiv.org/abs/2606.24292.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. VideoTree: Adaptive tree-based video representation for llm reasoning on long videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3272–3283. IEEE, 2025c.

Ziyang Wang, Honglu Zhou, Shijie Wang, Junnan Li, Caiming Xiong, Silvio Savarese, Mohit Bansal, Michael S. Ryoo, and Juan Carlos Niebles. Active Video Perception: Iterative evidence seeking for agentic long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 9088–9099, June 2026e.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13084–13094. IEEE, 2024.

xAI. RealWorldQA: A benchmark for real-world understanding. https://huggingface.co/datasets/xai-org/RealworldQA, 2024. Dataset benchmark released by xAI.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. NExT-QA: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9777–9786, June 2021.

Zhenghao Xing, Ruiyang Xu, Yuxuan Wang, Jinzheng He, Ziyang Ma, Qize Yang, Yunfei Chu, Jin Xu, Junyang Lin, Chi-Wing Fu, and Pheng-Ann Heng. Native active perception as reasoning for omni-modal understanding, 2026. URL https://arxiv.org/abs/2606.19341.

Sosuke Yamao, Natsuki Miyahara, Yuankai Qi, and Shun Takeuchi. Question-guided visual compression with memory feedback for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 32850–32859, June 2026.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. VisionZip: Longer is better but not necessary in vision language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802. IEEE, 2025a.

Senqiao Yang, Junyi Li, Xin Lai, Jinming Wu, Wei Li, Zejun MA, Bei Yu, Hengshuang Zhao, and Jiaya Jia. VisionThink: Smart and efficient vision language model via reinforcement learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 95187–95227. Curran Associates, Inc., 2025b. doi: 10.52202/085713-3182. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/88be023075a5a3ff3dc3b5d26623fa22-Paper-Conference.pdf.

Xinlei Yu, Chengming Xu, Guibin Zhang, Zhangquan Chen, Yudong Zhang, Yongbo He, Peng-Tao Jiang, Jiangning Zhang, Xiaobin Hu, and Shuicheng Yan. VisMem: Latent vision memory unlocks potential of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 31544–31555, June 2026.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. ActivityNet-QA: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 9127–9134, 2019.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9556–9567, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11975–11986, October 2023.

Jiarui Zhang, Muzi Tao, Shangshang Wang, Ollie Liu, Xuezhe Ma, and Willie Neiswanger. An exam for active observers. *arXiv preprint arXiv:2607.16165*, 2026.

Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, et al. Adaptive chain-of-focus reasoning via dynamic visual search and zooming for efficient vlms. *arXiv preprint arXiv:2505.15436*, 2025.

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

Kesen Zhao, Beier Zhu, Qianru Sun, and Hanwang Zhang. Unsupervised visual chain-of-thought reasoning via preference optimization. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2303–2312. IEEE, 2025.

Ziwei Zheng, Minghao Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and XingYu. DeepEyes: Incentivizing "thinking with images" via reinforcement learning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 126775–126798, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/cdb347d7516d52a9280ea9d6708911f6-Paper-Conference.pdf.

Muzhi Zhu, Hao Zhong, Canyu Zhao, Zongze Du, Mingyu Liu, Zheng Huang, Anzhou Li, Hao Chen, Cheng Zou, Jingdong Chen, Ming Yang, and Chunhua Shen. ACTIVE-o3 : Empowering MLLMs with active perception via pure reinforcement learning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=1uU7X9zzJB.

APPENDIX CONTENTS

A Notation and Indexing Conventions **17**

B Extended Budgeted Formulation and Theoretical Details **18**

B.1 Selective-Observation Control Process 18

B.2 Default Cost Instantiation 18

B.3 Feasible Views and Bellman Recursion 19

B.4 Proof of Proposition 1 20

B.5 Vector-Budget Extension 20

C Controller Parameterization and Memory Compression **21**

C.1 Shared Evidence State and Resource Conditioning 21

C.2 Budget-Feasible View Policy 21

C.3 Candidate Representation and Contextualization 22

C.4 Capacity-Preserving Memory Update 23

C.5 Task-Conditioned Rate–Distortion Interpretation 25

D Reinforcement-Learning and Training Details **26**

D.1 Potential-Based Credit and Continuation Value 26

D.2 Offline Demonstration Warm-Start 27

D.3 Heterogeneous Policy Update 29

D.4 Budget-Conditioned Training Schedule 30

E Inference Protocol and Complexity Analysis **31**

E.1 Deterministic Budgeted Execution 31

E.2 Resource Accounting and Computational Complexity 32

F Experimental Details **34**

F.1 Backbones and Visual Interface 34

F.2 Benchmarks and Task Metrics 34

F.3 Controller Training Data 35

F.4 Observation Operator and Resource Budgets 36
F.5 Controller and Memory Configuration 38
F.6 Training Protocol 38
F.7 Baseline Implementations and Fairness 39
F.8 Inference and System Measurement 41
F.9 Statistical Protocol 42

G Full Benchmark Results **42**
G.1 Full Qwen2.5-VL-7B Results 42
G.2 Transfer to Qwen3-VL-8B 43
G.3 Seed Variability 44
G.4 Paired Confidence Intervals 44

H Budget and System Efficiency **44**
H.1 Observation-Budget Sweeps 45
H.2 Resource Accounting 45
H.3 Deployment Latency and Memory 46
H.4 Iso-Quality and Iso-Latency Comparisons 46
H.5 Interaction Between Observation and Memory Budgets 46

I Ablations and Training Diagnostics **47**
I.1 Component Ablations 47
I.2 Feasible Action Support 49
I.3 Budget Interpolation 49
I.4 Training Sample Efficiency and the Role of Demonstrations 49
I.5 Does Reinforcement Learning Alone Explain the Gain? 50
I.6 Resource-Regularization and Preview Sensitivity 51

J Theory-Aligned Analyses **52**
J.1 Counterfactual Nested Halting 52
J.2 Empirical Continuation Advantage 52
J.3 Query-Conditioned Rate–Distortion 53
J.4 Input-Adaptive Computation 54

K Qualitative Results and Failure Cases **55**
K.1 Qualitative Trajectories and Memory Dynamics 55
K.2 Failure Modes 55
K.3 OOD Stress Test on ActiveVision 56

APPENDIX

A NOTATION AND INDEXING CONVENTIONS

This appendix summarizes the core notation used throughout WATCH. We list only recurring quantities that define the model, resource constraints, and learning objective; local variables used only inside individual derivations are introduced where needed.

Indexing convention. Index k denotes a controller decision made from the current persistent memory. Conditional on continuing, the k-th acquisition produces visual tokens indexed by $k+1$:

$$(\mathbf{M}_k, \mathbf{L}_k) \xrightarrow{v_k} (\widetilde{\mathbf{Z}}_{k+1}, \widetilde{\mathbf{L}}_{k+1}) \longrightarrow (\mathbf{C}_{k+1}, \mathbf{L}^C_{k+1}) \xrightarrow{a_k^{\mathrm{mem}}} (\mathbf{M}_{k+1}, \mathbf{L}_{k+1}).$$

Thus $\mathbf{M}_k$ always denotes evidence available *before* the k-th adaptive acquisition, whereas quantities indexed by $k+1$ are created by or after that acquisition. If perception terminates after K acquisitions, $\mathbf{M}_K$ is the only visual memory passed to the language backbone.

The observation and memory budgets play different roles: B_{obs} is a cumulative acquisition budget with remaining amount b_k, whereas B_{mem} is a hard instantaneous capacity. Accordingly, $c_{\mathrm{mem}}(\mathbf{M}_K)$ is a soft terminal context cost and is not a cumulatively depleted memory budget.

Symbol	Meaning
$q, \mathcal{V}, x$	Text query, raw visual input, and multimodal input $x = (q, \mathcal{V})$.
E_v, π_ϕ	Frozen visual front-end and language backbone; the backbone receives only the terminal visual memory at inference.
$\mathbf{M}_k, \mathbf{L}_k$	Persistent backbone-compatible visual feature bank and its index-aligned controller-side spatial–temporal metadata bank.
$\widetilde{\mathbf{Z}}_{k+1}, \widetilde{\mathbf{L}}_{k+1}$	Visual features acquired by view v_k and their aligned metadata.
$\mathbf{C}_{k+1}, \mathbf{L}^C_{k+1}$	Candidate feature and metadata banks obtained by combining persistent and newly acquired evidence before memory control.
$\mathbf{h}_k, z_k$	Recurrent evidence summary and budget-independent evidence state $z_k = (\mathbf{h}_k, \mathbf{M}_k, \mathbf{L}_k)$.
B_{obs}, b_k	Total observation budget and observation budget remaining before decision k.
B_{mem}	Hard maximum number of visual representations retained in persistent memory.
$K_{\max}, K$	Maximum adaptive horizon and realized number of acquisitions before termination.
$\boldsymbol{\beta}_k, \mathbf{d}_k$	Normalized resource descriptor and budget-conditioned policy context.
a_k^{halt}	Binary decision to terminate perception and answer from the current memory.
$v_k = (g_k, I_k, R_k, \rho_k)$	Adaptive view action: observation mode, temporal support, spatial region, and fidelity.
$a_k^{\mathrm{mem}}, \mathcal{S}_{k+1}$	Autoregressive memory action and resulting set of retained anchor indices.
$J_{k+1}, R_k^{\mathrm{mem}}$	Number of retained anchors and number of realized stochastic memory-policy rounds, including STOP when termination occurs before capacity.
$c_{\mathrm{obs}}, c_{\mathrm{mem}}$	Per-view acquisition cost and terminal visual-context cost.
$Q(x, y), R_g$	Final task-quality score and resource-regularized terminal utility of rollout g.
$\mathcal{F}_{\mathrm{view}}(z, b)$	Set of visual views feasible under evidence state z and remaining observation budget b.
$H(z), V_\ell^\star(z, b)$	Utility of halting immediately and optimal finite-horizon value with ℓ acquisition opportunities remaining.
θ, ξ	WATCH policy parameters and disjoint training-time potential parameters.
$\bar{\zeta}^M, \bar{\zeta}^C$	Policy-parameter-independent pre-acquisition and post-acquisition shaping state representations used for dense credit assignment.
Φ_ξ	Training-only continuation-value potential used for policy-invariant dense credit.
$G^d_{g,k}, A^d_{g,k}$	Realized continuation return and group-relative advantage for controller stage $d \in \{\mathrm{halt}, \mathrm{view}, \mathrm{mem}\}$.

B Extended Budgeted Formulation and Theoretical Details

This appendix formalizes the selective-observation process underlying WATCH, specifies the default cost normalization, derives the finite-horizon Bellman recursion in Section 3.1, and proves Proposition 1. We finally give a vector-budget extension for deployment settings with multiple cumulative resources.

B.1 Selective-Observation Control Process

WATCH can be viewed as a selective-observation POMDP in which the raw visual input exists in the environment but is not processed exhaustively. The partial observability is therefore computational: task-relevant evidence becomes available to the controller only through explicit, costly observations.

For a fixed multimodal input $x = (q, \mathcal{V})$, let $\mathcal{H}_k$ denote the complete history before the k-th adaptive decision, including the initial preview, previously selected views, acquired visual evidence, and memory updates. An exact information state is any sufficient statistic of $\mathcal{H}_k$ for predicting subsequent observations and utilities. WATCH approximates such an information state by

$$z_k = (\mathbf{h}_k, \mathbf{M}_k, \mathbf{L}_k),$$

where $\mathbf{M}_k$ is the persistent backbone-compatible visual feature bank, $\mathbf{L}_k$ its aligned controller-side metadata bank, and $\mathbf{h}_k$ a learned recurrent evidence summary. Importantly, the current resource budget is not part of z_k; it is supplied separately to the policy through the resource descriptor $\boldsymbol{\beta}_k$.

The initialization is policy independent. Specifically,

$$(\mathbf{M}_0, \mathbf{L}_0) = \mathcal{P}_0\big(E_v(\mathcal{O}_0(\mathcal{V})); B_{\mathrm{mem}}\big),$$

where both $\mathcal{O}_0$ and $\mathcal{P}_0$ are deterministic and contain no trainable WATCH parameters. Assuming $B_{\mathrm{mem}} \geq 1$, $\mathcal{P}_0$ uses fixed pooling when necessary and guarantees $1 \leq |\mathbf{M}_0| = |\mathbf{L}_0| \leq B_{\mathrm{mem}}$. Consequently, for a fixed input and prescribed budget, $(\mathbf{M}_0, \mathbf{L}_0, b_0)$ is identical across policy rollouts.

At a non-terminal evidence state z_k, the controller first selects either the absorbing halt action $\dagger$ or a visual view v_k. Choosing $\dagger$ invokes the language backbone once on $\mathbf{M}_k$ and terminates the trajectory. A continuation view produces

$$\widetilde{\mathcal{V}}_{k+1} = \mathcal{O}(\mathcal{V}, v_k), \qquad \widetilde{\mathbf{Z}}_{k+1} = E_v(\widetilde{\mathcal{V}}_{k+1}), \qquad \widetilde{\mathbf{L}}_{k+1} = \mathrm{Meta}(v_k, \widetilde{\mathbf{Z}}_{k+1}). \tag{12}$$

For compactness, write $\widetilde{\mathcal{E}} = (\widetilde{\mathbf{Z}}, \widetilde{\mathbf{L}})$ for the feature–metadata observation pair, and denote its observation kernel by $\mathcal{G}\big(\mathrm{d}\widetilde{\mathcal{E}} \mid z, v\big)$. For deterministic E_v, $\mathcal{O}$, and Meta, this kernel reduces to a point mass.

Crucially, memory control occurs after $\widetilde{\mathcal{E}}$ is observed. Let $\mathcal{M}\big(z, \widetilde{\mathcal{E}}; B_{\mathrm{mem}}\big)$ denote the feasible memory-action set. For $a^{\mathrm{mem}} \in \mathcal{M}(z, \widetilde{\mathcal{E}}; B_{\mathrm{mem}})$,

$$(\mathbf{M}', \mathbf{L}') = \mathcal{K}\big(\mathbf{M}, \mathbf{L}, \widetilde{\mathcal{E}}, a^{\mathrm{mem}}\big), \qquad 1 \leq |\mathbf{M}'| = |\mathbf{L}'| \leq B_{\mathrm{mem}},$$

and

$$z' = \mathcal{T}\big(z, \widetilde{\mathcal{E}}, a^{\mathrm{mem}}\big).$$

We therefore define the admissible post-observation evidence set

$$\Gamma_{B_{\mathrm{mem}}}(z, \widetilde{\mathcal{E}}) = \Big\{ \mathcal{T}(z, \widetilde{\mathcal{E}}, a^{\mathrm{mem}}) : a^{\mathrm{mem}} \in \mathcal{M}(z, \widetilde{\mathcal{E}}; B_{\mathrm{mem}}) \Big\}. \tag{13}$$

The concrete implementation of $\mathcal{K}$ and the memory action space is given in Section 3.2 and Appendix C.

B.2 Default Cost Instantiation

The formulation permits deployment-specific cost measurements, but we use fixed normalized proxies as the default training costs.

For a view $v = (g, I, R, \rho)$, where I is the selected temporal support, R the normalized spatial region, and ρ the linear fidelity scale, we define

$$c_{\mathrm{obs}}(v) = \frac{\alpha_{\mathrm{enc}} + \alpha_{\mathrm{pix}}|I|\,\mathrm{Area}(R)\rho^2}{\alpha_{\mathrm{enc}} + \alpha_{\mathrm{pix}}}, \tag{14}$$

where $\alpha_{\mathrm{enc}} \geq 0$ and $\alpha_{\mathrm{pix}} > 0$ control the relative fixed encoding and pixel-processing costs. The normalization assigns unit cost to a single native-fidelity full-frame observation. The preview cost is $c_{\mathrm{pre}} = c_{\mathrm{obs}}(v_{\mathrm{pre}})$ for the fixed preview configuration.

For the terminal memory, we use

$$c_{\mathrm{mem}}(\mathbf{M}) = \frac{|\mathbf{M}|}{B_{\mathrm{mem}}^{\max}}, \tag{15}$$

where $B_{\mathrm{mem}}^{\max}$ is the largest supported memory capacity. This normalization preserves the ordering of absolute terminal context sizes across different operating budgets. Actual visual tokens, FLOPs, and measured latency are reported separately as deployment metrics rather than substituted for the training proxy; see Appendix E.

B.3 Feasible Views and Bellman Recursion

Let $b \geq 0$ denote the remaining observation budget. The feasible view set is

$$\mathcal{F}_{\mathrm{view}}(z, b) = \{v : c_{\mathrm{obs}}(v) \leq b\}. \tag{16}$$

The actual controller further parameterizes this set so that only views with a feasible continuation are sampled, as detailed in Appendix C.2.

For compactness, define the utility of halting at $z = (\mathbf{h}, \mathbf{M}, \mathbf{L})$ as

$$H(z) = \mathbb{E}_{y \sim \pi_\phi(\cdot|q,\mathbf{M})}\left[Q(x, y)\right] - \lambda_{\mathrm{mem}} c_{\mathrm{mem}}(\mathbf{M}). \tag{17}$$

For fixed B_{mem}, define the optimal continuation value with $\ell \geq 1$ remaining acquisitions as

$$\begin{aligned} C_\ell^\star(z, b) = \sup_{v \in \mathcal{F}_{\mathrm{view}}(z,b)} \Bigg[& -\lambda_{\mathrm{obs}} c_{\mathrm{obs}}(v) \\ & + \int \sup_{z' \in \Gamma_{B_{\mathrm{mem}}}(z, \widetilde{\mathcal{E}})} V_{\ell-1}^\star\big(z', b - c_{\mathrm{obs}}(v)\big)\, \mathcal{G}(\mathrm{d}\widetilde{\mathcal{E}} \mid z, v) \Bigg]. \end{aligned} \tag{18}$$

The Bellman recursion is therefore

$$V_\ell^\star(z, b) = \max\left\{H(z), C_\ell^\star(z, b)\right\}, \qquad V_0^\star(z, b) = H(z). \tag{19}$$

If $\mathcal{F}_{\mathrm{view}}(z, b) = \varnothing$, the supremum in Eq. 18 is defined as $-\infty$, so halting is selected automatically.

This recursion distinguishes two forms of termination. Halting is *endogenous* when the current evidence already dominates every affordable continuation, and *budget forced* when no admissible view remains.

Policy evaluation and the implemented action class. For an exact information state s (the sufficient statistic in Appendix B.1, or the complete observation history), fix the controller, terminal decoder, and memory transition. Let $\alpha_\theta(s, b, \ell)$ be its halting probability and let $\mathcal{T}(s, v, \widetilde{E}, a)$ update the information state after acquisition and retention. The fixed-policy value satisfies

$$\begin{aligned} V_\ell^\theta(s, b) &= \alpha_\theta(s, b, \ell) H(s) + (1 - \alpha_\theta(s, b, \ell))\, \mathbb{E}_{v, \widetilde{E}, a}\Big[-\lambda_{\mathrm{obs}} c_{\mathrm{obs}}(v) + V_{\ell-1}^\theta\Big(\mathcal{T}(s, v, \widetilde{E}, a), b - c_{\mathrm{obs}}(v)\Big)\Big], \\ V_0^\theta(s, b) &= H(s). \end{aligned} \tag{20}$$

Here v is sampled from the feasible view policy, $\widetilde{E}$ from the observation kernel, and a from the post-acquisition memory policy; halting is forced when no view is feasible. The terminal utility H uses the fixed decoder as in Eq. 17. WATCH represents s by its recurrent summary $z = (h, M, L)$. For candidate bank C, its memory actions are ordered distinct anchor sequences of length $1 \leq m \leq \min\{B_{\mathrm{mem}}, |C|\}$, followed by the geometry-constrained merge map. These actions instantiate the feasible set in Eq. 13; its induced $\Gamma_{B_{\mathrm{mem}}}$ contains exactly their reachable memory updates. Maximizing over the supported actions gives optimal control within this action class, whereas Eq. 20 evaluates a fixed policy. The learned heads jointly approximate its stopping and continuation decisions through the rollout objective.

B.4 Proof of Proposition 1

Proof. We prove budget monotonicity by induction on the remaining horizon ℓ.

For $\ell = 0$,

$$V_0^\star(z, b) = H(z)$$

for every b, so the claim is immediate.

Assume that for every evidence state z',

$$V_{\ell-1}^\star(z', b_1) \le V_{\ell-1}^\star(z', b_2) \qquad \text{whenever } b_1 \le b_2.$$

Consider $b_1 \le b_2$. By assumption,

$$\mathcal{F}_{\text{view}}(z, b_1) \subseteq \mathcal{F}_{\text{view}}(z, b_2).$$

Take any $v \in \mathcal{F}_{\text{view}}(z, b_1)$. The same view is feasible under b_2, and

$$b_1 - c_{\text{obs}}(v) \le b_2 - c_{\text{obs}}(v).$$

For every realized $\widetilde{\mathcal{E}}$ and every $z' \in \Gamma_{B_{\text{mem}}}(z, \widetilde{\mathcal{E}})$, the induction hypothesis gives

$$V_{\ell-1}^\star\big(z', b_1 - c_{\text{obs}}(v)\big) \le V_{\ell-1}^\star\big(z', b_2 - c_{\text{obs}}(v)\big).$$

Taking the supremum over the same admissible post-observation states, integrating with respect to the same observation kernel, and finally taking the supremum over the nested view sets preserves the inequality. Hence

$$C_\ell^\star(z, b_1) \le C_\ell^\star(z, b_2).$$

Because $H(z)$ does not depend on the remaining observation budget,

$$\begin{aligned} V_\ell^\star(z, b_1) &= \max\{H(z), C_\ell^\star(z, b_1)\} \\ &\le \max\{H(z), C_\ell^\star(z, b_2)\} \\ &= V_\ell^\star(z, b_2). \end{aligned}$$

This proves the first claim.

Now suppose halting is optimal at (z, b_2). Then

$$H(z) \ge C_\ell^\star(z, b_2).$$

Since $C_\ell^\star(z, b_1) \le C_\ell^\star(z, b_2)$ for every $b_1 \le b_2$,

$$H(z) \ge C_\ell^\star(z, b_1),$$

so halting is also optimal at (z, b_1). Therefore the halting region is nested and expands as the available observation budget decreases. □

B.5 Vector-Budget Extension

The main formulation isolates observation cost and instantaneous memory capacity because they correspond directly to WATCH's acquisition and retention interfaces. The same analysis extends to multiple cumulative resources whose per-view costs are known, or conservatively upper-bounded, when a decision is made.

Let

$$\mathbf{b} = (b^{(1)}, \dots, b^{(m)}) \in \mathbb{R}_+^m, \qquad \mathbf{c}(v) \in \mathbb{R}_+^m,$$

and define

$$\mathcal{F}_{\text{view}}(z, \mathbf{b}) = \{v : \mathbf{c}(v) \preceq \mathbf{b}\}.$$

The continuation recursion becomes

$$\begin{aligned} V_\ell^\star(z, \mathbf{b}) = \max\Bigg\{ H(z), \sup_{v \in \mathcal{F}_{\text{view}}(z, \mathbf{b})} \Big[-\boldsymbol{\lambda}^\top \mathbf{c}(v) \\ + \mathbb{E}_{\widetilde{\mathcal{E}}|z,v}\Big[\sup_{z' \in \Gamma_{B_{\text{mem}}}(z, \widetilde{\mathcal{E}})} V_{\ell-1}^\star\big(z', \mathbf{b} - \mathbf{c}(v)\big) \Big] \Big] \Bigg\}. \end{aligned} \tag{21}$$

Under the componentwise order $\mathbf{b}_1 \preceq \mathbf{b}_2$, the proof of Proposition 1 carries over whenever the feasible view sets are nested accordingly. This allows additional predictable resources—such as visual-token, energy, or latency budgets—to be incorporated without changing the Acquire–Retain–Halt structure.

C Controller Parameterization and Memory Compression

This appendix provides implementation details for the Acquire–Retain–Halt controller in Section 3.2. We specify the budget-independent recurrent evidence state, budget-feasible mixed view policy, metadata-aware candidate contextualization, capacity-constrained sequential anchor selection, geometry-aware redundancy resolution, and the task-conditioned rate–distortion interpretation.

C.1 Shared Evidence State and Resource Conditioning

The query is encoded once as $\bar{\mathbf{q}} = E_q(q) \in \mathbb{R}^{d_q}$. WATCH stores visual features and controller-side metadata in two aligned banks. The persistent visual memory is $\mathbf{M}_k = \{\mathbf{m}_{k,i}\}_{i=1}^{N_k}$, where each $\mathbf{m}_{k,i}$ is backbone compatible, while $\mathbf{L}_k = \{\boldsymbol{\ell}_{k,i}\}_{i=1}^{N_k}$ stores the corresponding normalized spatial–temporal metadata. The two banks are index aligned and satisfy $|\mathbf{M}_k| = |\mathbf{L}_k| = N_k$. Only M_k is exposed to the backbone as visual features; L_k remains controller-side.

For pooling, we construct $\mathbf{a}_{k,i} = [P_h(\mathbf{m}_{k,i}); E_\ell(\boldsymbol{\ell}_{k,i})]$, and obtain $\mathbf{m}_k^{\mathrm{pool}} = \mathrm{AttnPool}\left(\{\mathbf{a}_{k,i}\}_{i=1}^{N_k}\right)$.

Thus the recurrent state can summarize not only what evidence is stored but also where and at what scale it was acquired.

The budget-independent evidence recurrence is

$$\mathbf{h}_k = \mathrm{GRU}_\theta\left(\mathbf{h}_{k-1}, [\bar{\mathbf{q}}; \mathbf{m}_k^{\mathrm{pool}}]\right), \tag{22}$$

although a shallow recurrent Transformer may be substituted without changing the formulation.

The resource descriptor is kept separate from this recurrence and follows Section 3.1:

$$\boldsymbol{\beta}_k = \left(\frac{b_k}{B_{\mathrm{obs}}^{\max}}, \frac{B_{\mathrm{obs}}}{B_{\mathrm{obs}}^{\max}}, \frac{B_{\mathrm{mem}}}{B_{\mathrm{mem}}^{\max}}, \frac{|\mathbf{M}_k|}{B_{\mathrm{mem}}}, \frac{K_{\max} - k}{K_{\max}}\right), \qquad \mathbf{d}_k = [\mathbf{h}_k; E_\beta(\boldsymbol{\beta}_k)]. \tag{23}$$

Consequently, the recurrent evidence state remains unchanged when only the deployment budget changes, while all policy heads can still adapt to the requested operating point.

After executing the k-th view, the memory decision uses a post-acquisition resource descriptor

$$\boldsymbol{\beta}_k^+ = \left(\frac{b_{k+1}}{B_{\mathrm{obs}}^{\max}}, \frac{B_{\mathrm{obs}}}{B_{\mathrm{obs}}^{\max}}, \frac{B_{\mathrm{mem}}}{B_{\mathrm{mem}}^{\max}}, \frac{|\mathbf{M}_k|}{B_{\mathrm{mem}}}, \frac{K_{\max} - k - 1}{K_{\max}}\right), \qquad \mathbf{d}_k^+ = [\mathbf{h}_k; E_\beta(\boldsymbol{\beta}_k^+)]. \tag{24}$$

The candidate-bank size itself is directly observed by the token contextualizer and therefore need not be redundantly included in $\boldsymbol{\beta}_k^+$.

The initial aligned banks $(\mathbf{M}_0, \mathbf{L}_0)$ are constructed deterministically by $\mathcal{P}_0$ in Section 3.1; the learned memory controller is not used during initialization. A learned constant $\mathbf{h}_{-1}$ is then combined with the aligned preview banks $(\mathbf{M}_0, \mathbf{L}_0)$ through Eq. 22.

C.2 Budget-Feasible View Policy

The canonical view action is $v_k = (g_k, I_k, R_k, \rho_k)$, where $g_k \in \{\mathrm{global}, \mathrm{local}\}$ selects the observation mode, I_k specifies temporal support, R_k a spatial region, and ρ_k the linear fidelity scale. For a single image, $I_k = \{1\}$; for video, the admissible temporal set may contain individual frames, short contiguous windows, or sparsely sampled frame groups.

We use the autoregressive factorization

$$\begin{aligned} \pi_\theta^{\mathrm{view}}(v_k \mid \mathbf{d}_k) &= \pi_\theta^g(g_k \mid \mathbf{d}_k)\, \pi_\theta^I(I_k \mid g_k, \mathbf{d}_k) \\ &\times \pi_\theta^R(R_k \mid g_k, I_k, \mathbf{d}_k, b_k)\, \pi_\theta^\rho(\rho_k \mid g_k, I_k, R_k, \mathbf{d}_k, b_k). \end{aligned} \tag{25}$$

For $g_k = \mathrm{global}$, π_θ^R degenerates to a point mass at R_{full}.

Hard budget compliance is enforced at every prefix of this factorization. Let

$$A_{\min}^{\mathrm{local}} = w_{\min} h_{\min}, \qquad A_{\min}^{\mathrm{global}} = 1.$$

Given the default cost in Eq. 14, a pair (g, I) admits at least one feasible completion whenever

$$c_{\text{obs}}\left(g, I, R^{g}_{\min}, \rho_{\min}\right) \le b_k, \tag{26}$$

where $R^{\text{global}}_{\min} = R_{\text{full}}$ and $R^{\text{local}}_{\min}$ denotes any minimum-area local box. A mode g is masked if no temporal choice satisfies Eq. 26; conditional on the selected mode, all infeasible I are masked before sampling. Hence every sampled discrete prefix has at least one feasible continuous completion.

For a local view, define the maximum area that remains feasible at the minimum fidelity:

$$A^{\text{feas}}_{\max}(I, b) = \min\left\{1,\ \frac{[b(\alpha_{\text{enc}} + \alpha_{\text{pix}}) - \alpha_{\text{enc}}]_+}{\alpha_{\text{pix}}|I|\rho^2_{\min}}\right\}, \tag{27}$$

where $[x]_+ = \max\{x, 0\}$ and $\alpha_{\text{pix}} > 0$ for the default parameterization. Feasibility masking ensures $A^{\text{feas}}_{\max} \ge w_{\min} h_{\min}$.

We sample the local width from

$$w^{\text{box}}_k = w_{\min} + \upsilon^w_k \left(w^{\text{feas}}_{\max} - w_{\min}\right), \qquad \upsilon^w_k \sim \text{Beta}(\alpha^w_k, \beta^w_k),$$

where

$$w^{\text{feas}}_{\max} = \min\left\{w_{\max},\ \frac{A^{\text{feas}}_{\max}(I_k, b_k)}{h_{\min}}\right\}.$$

Conditional on this width,

$$h^{\text{feas}}_{\max} = \min\left\{h_{\max},\ \frac{A^{\text{feas}}_{\max}(I_k, b_k)}{w^{\text{box}}_k}\right\},$$

and

$$h^{\text{box}}_k = h_{\min} + \upsilon^h_k \left(h^{\text{feas}}_{\max} - h_{\min}\right), \qquad \upsilon^h_k \sim \text{Beta}(\alpha^h_k, \beta^h_k).$$

The box location is generated by

$$x_k = (1 - w^{\text{box}}_k)\upsilon^x_k, \qquad y_k = (1 - h^{\text{box}}_k)\upsilon^y_k,$$

where

$$\upsilon^x_k \sim \text{Beta}(\alpha^x_k, \beta^x_k), \qquad \upsilon^y_k \sim \text{Beta}(\alpha^y_k, \beta^y_k).$$

Therefore

$$R_k = \left(x_k, y_k, w^{\text{box}}_k, h^{\text{box}}_k\right) \subseteq [0, 1]^2$$

and $\text{Area}(R_k) \le A^{\text{feas}}_{\max}(I_k, b_k)$ by construction.

After I_k and R_k are fixed, the largest admissible fidelity is

$$\rho^{\text{feas}}_{\max}(I, R, b) = \min\left\{\rho_{\max},\ \sqrt{\frac{[b(\alpha_{\text{enc}} + \alpha_{\text{pix}}) - \alpha_{\text{enc}}]_+}{\alpha_{\text{pix}}|I|\,\text{Area}(R)}}\right\}. \tag{28}$$

The fidelity is then sampled as

$$\rho_k = \rho_{\min} + \upsilon^\rho_k \left(\rho^{\text{feas}}_{\max}(I_k, R_k, b_k) - \rho_{\min}\right), \qquad \upsilon^\rho_k \sim \text{Beta}(\alpha^\rho_k, \beta^\rho_k). \tag{29}$$

The same prefix-feasibility rules are used during training and inference, ensuring that every executed view is feasible without a deployment-time projection.

C.3 Candidate Representation and Contextualization

The memory controller operates on lightweight projections rather than the full backbone features. Each candidate token is mapped by

$$P_c : \mathbb{R}^{d_v} \to \mathbb{R}^{d_c}, \qquad d_c \ll d_v,$$

where P_c is a linear projection or a shallow MLP trained jointly with the controller.

The aligned candidate metadata bank introduced in Section 3.2 is

$$\mathbf{L}^C_{k+1} = \left[\mathbf{L}_k; \widetilde{\mathbf{L}}_{k+1}\right] = \{\boldsymbol{\ell}_{k+1,i}\}_{i=1}^{N^c_{k+1}}.$$

Each metadata vector is parameterized as

$$\boldsymbol{\ell}_{k+1,i} = (t_{k+1,i}, x_{k+1,i}, y_{k+1,i}, \rho_{k+1,i}), \tag{30}$$

describing temporal position, global spatial position, and acquisition fidelity. Coordinates produced inside a crop are mapped back to the original visual coordinate system before being stored. For single-image inputs, the temporal coordinate is constant.

A binary source indicator $\chi_{k+1,i} \in \{\text{old}, \text{new}\}$ distinguishes tokens inherited from $\mathbf{M}_k$ from tokens in $\widetilde{\mathbf{Z}}_{k+1}$. Source is defined relative to the current transition; once $\mathbf{M}_{k+1}$ is formed, all of its elements are treated as inherited evidence at the next step.

For newly acquired tokens, novelty is computed by Eq. 5. Each candidate is represented to the memory controller as

$$\mathbf{r}_{k+1,i} = [P_c(\mathbf{c}_{k+1,i}); E_\ell(\boldsymbol{\ell}_{k+1,i}); E_\chi(\chi_{k+1,i}); \nu_{k+1,i}]. \tag{31}$$

A shallow L_m-layer self-attention module then contextualizes the complete candidate bank:

$$\{\mathbf{u}_{k+1,i}\}_{i=1}^{N_{k+1}^c} = G_\theta\left(\{\mathbf{r}_{k+1,i}\}_{i=1}^{N_{k+1}^c}, \mathbf{d}_k^+\right). \tag{32}$$

Because candidates interact inside G_θ, their representations can jointly encode query-conditioned relevance, spatial–temporal coverage, novelty, and redundancy. These contextualized representations parameterize the capacity-constrained anchor policy described next.

C.4 Capacity-Preserving Memory Update

Capacity-constrained anchor selection. Let $B_{k+1}^a = \min\{B_{\text{mem}}, N_{k+1}^c\}$ be the largest admissible anchor count for the current candidate bank. The memory action is generated autoregressively without replacement.

Let $\mathcal{S}_{k+1}^{(0)} = \varnothing$. At selection round r, we summarize the anchors already selected by

$$\mathbf{q}_{k+1}^{(r)} = H_\theta\left(\mathbf{d}_k^+, \text{Pool}_a\left(\{\mathbf{u}_{k+1,i} : i \in \mathcal{S}_{k+1}^{(r-1)}\}\right), E_r\left(\frac{r}{B_{k+1}^a}\right)\right), \tag{33}$$

where $\text{Pool}_a(\varnothing)$ is a learned null embedding. For every unselected candidate $i \notin \mathcal{S}_{k+1}^{(r-1)}$, the selector assigns the logit

$$\ell_{k+1,i}^{(r)} = \mathbf{w}_a^\top\left[\mathbf{u}_{k+1,i}; \mathbf{q}_{k+1}^{(r)}\right] + b_a,$$

while the stopping logit is

$$\ell_{k+1,\text{stop}}^{(r)} = \mathbf{w}_s^\top \mathbf{q}_{k+1}^{(r)} + b_s.$$

The r-th anchor action is sampled from

$$a_{k+1,r}^{\text{anc}} \sim \text{Cat}\left(\text{softmax}\left(\{\ell_{k+1,i}^{(r)} : i \notin \mathcal{S}_{k+1}^{(r-1)}\} \cup \{\ell_{k+1,\text{stop}}^{(r)}\}\right)\right). \tag{34}$$

At $r = 1$, the STOP action is masked, guaranteeing at least one retained anchor. If a candidate i is selected, we set $\mathcal{S}_{k+1}^{(r)} = \mathcal{S}_{k+1}^{(r-1)} \cup \{i\}$ and continue. Selection terminates when STOP is sampled or when $r = B_{k+1}^a$.

Let $J_{k+1} = |\mathcal{S}_{k+1}|$ denote the number of selected anchors. The complete stochastic memory action includes the terminating STOP decision whenever selection ends before reaching capacity:

$$a_k^{\text{mem}} = \begin{cases} \left(i_{k+1,1}, \ldots, i_{k+1,J_{k+1}}, \text{STOP}\right), & J_{k+1} < B_{k+1}^a, \\ \left(i_{k+1,1}, \ldots, i_{k+1,B_{k+1}^a}\right), & J_{k+1} = B_{k+1}^a. \end{cases} \tag{35}$$

Accordingly, the number of realized stochastic memory-policy rounds is

$$R_k^{\text{mem}} = J_{k+1} + \mathbf{1}\left[J_{k+1} < B_{k+1}^a\right]. \tag{36}$$

The associated anchor set contains only selected candidate indices,

$$\mathcal{S}_{k+1} = \left\{i_{k+1,1}, \ldots, i_{k+1,J_{k+1}}\right\}, \qquad 1 \le J_{k+1} \le B_{k+1}^a \le B_{\text{mem}}.$$

Thus STOP is part of the stochastic action likelihood but never part of the retained anchor set. The ordering and optional stopping decision are used to factorize the memory policy; the subsequent deterministic memory transition depends only on the completed anchor set.

At inference time, categorical sampling is replaced by greedy decoding: WATCH repeatedly selects the highest-probability admissible candidate or STOP, with STOP disabled at the first round, until stopping or reaching B^a_{k+1} anchors.

Geometry-aware redundancy resolution. Anchor selection determines which candidates remain explicitly represented, but not every non-anchor candidate should be discarded. We merge a non-anchor token only when it is both semantically redundant with and geometrically compatible with a retained anchor.

For candidates i and j, define their semantic similarity as

$$s^{\mathrm{sem}}_{ij} = \cos\left(P_c(\mathbf{c}_{k+1,i}), P_c(\mathbf{c}_{k+1,j})\right). \tag{37}$$

Using $\boldsymbol{\ell}_{k+1,i} = (t_i, x_i, y_i, \rho_i)$, we define the normalized geometric discrepancy

$$d^{\mathrm{geo}}_{ij} = \omega_{xy} \left\|(x_i, y_i) - (x_j, y_j)\right\|_2 + \omega_t |t_i - t_j| + \omega_\rho \left|\log(\rho_i + \epsilon) - \log(\rho_j + \epsilon)\right|, \tag{38}$$

where $\omega_{xy}, \omega_t, \omega_\rho \geq 0$. For single-image inputs, $\omega_t = 0$.

For every non-anchor candidate $j \notin \mathcal{S}_{k+1}$, define the compatible-anchor set

$$\mathcal{S}^{\mathrm{comp}}_{k+1}(j) = \left\{ i \in \mathcal{S}_{k+1} : s^{\mathrm{sem}}_{ij} \geq \tau_{\mathrm{sem}}, \quad d^{\mathrm{geo}}_{ij} \leq \tau_{\mathrm{geo}} \right\}. \tag{39}$$

If $\mathcal{S}^{\mathrm{comp}}_{k+1}(j) = \varnothing$, candidate j is pruned. Otherwise it is assigned to the most semantically similar compatible anchor,

$$\mu_{k+1}(j) = \arg \max_{i \in \mathcal{S}^{\mathrm{comp}}_{k+1}(j)} s^{\mathrm{sem}}_{ij}. \tag{40}$$

Let $\mathcal{A}_{k+1,i}$ denote the non-anchor candidates assigned to anchor i and define $\mathcal{G}_{k+1,i} = \{i\} \cup \mathcal{A}_{k+1,i}$. The merged backbone-space representation is

$$\mathbf{m}_{k+1,i} = \sum_{j \in \mathcal{G}_{k+1,i}} \alpha_{ij} \mathbf{c}_{k+1,j}, \qquad \alpha_{ij} = \frac{\exp\left(s^{\mathrm{sem}}_{ij}/\tau_m\right)}{\sum_{r \in \mathcal{G}_{k+1,i}} \exp\left(s^{\mathrm{sem}}_{ir}/\tau_m\right)}, \tag{41}$$

where $\tau_m > 0$. The aggregation weights are therefore nonnegative and sum to one.

Unlike the visual representation, controller-side metadata are not averaged. Because merging is restricted to geometrically compatible evidence, the merged token inherits the metadata of its retained anchor:

$$\boldsymbol{\ell}^{\mathrm{merge}}_{k+1,i} = \boldsymbol{\ell}_{k+1,i}. \tag{42}$$

Consequently, a merged feature also inherits the backbone visual position indices associated with its retained anchor. This avoids creating artificial spatial or temporal locations that do not correspond to any actually observed token.

The update produces two index-aligned banks,

$$\mathbf{M}_{k+1} = \left\{ \mathbf{m}_{k+1,i} : i \in \mathcal{S}_{k+1} \right\}, \qquad \mathbf{L}_{k+1} = \left\{ \boldsymbol{\ell}_{k+1,i} : i \in \mathcal{S}_{k+1} \right\}, \tag{43}$$

with $|\mathbf{M}_{k+1}| = |\mathbf{L}_{k+1}|$. Only $\mathbf{M}_{k+1}$ is exposed to the backbone; $\mathbf{L}_{k+1}$ remains controller-side state for subsequent WATCH decisions.

Here E_v denotes the complete visual front-end up to the backbone-compatible visual-token interface, so merging is performed in that interface space. If a particular backbone does not support feature-space token merging, WATCH disables the merge operator and interprets all non-anchor candidates as pruned. Thus merging improves redundancy resolution when compatible with the backbone but is not required by the overall WATCH framework.

The capacity-preserving memory transition is illustrated in Figure 3.

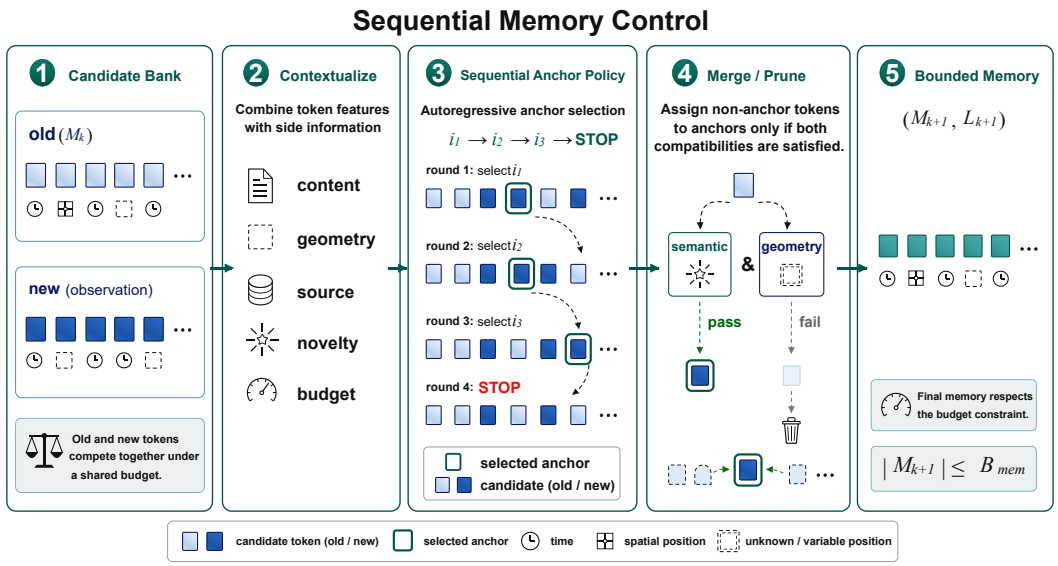


Figure 3: Visualization of the WATCH memory controller. Candidate tokens from previous memory and newly acquired observations are jointly contextualized, selected through a capacity-constrained sequential anchor policy, and resolved by geometry-aware redundancy handling.

C.5 Task-Conditioned Rate–Distortion Interpretation

The memory update admits a task-conditioned rate–distortion interpretation. For candidate evidence $\mathbf{C}$ and compressed memory $\mathbf{M}$ that are valid backbone-compatible contexts, define

$$P_{\mathbf{C}}(y) = \pi_\phi(y \mid q, \mathbf{C}), \qquad P_{\mathbf{M}}(y) = \pi_\phi(y \mid q, \mathbf{M}).$$

The task-conditioned distortion is

$$D_q(\mathbf{C}, \mathbf{M}) = D_{\mathrm{KL}}\left(P_{\mathbf{C}} \,\|\, P_{\mathbf{M}}\right). \tag{44}$$

Using the same terminal context cost $c_{\mathrm{mem}}(\mathbf{M})$ as the main objective, an ideal compressor would solve

$$\mathbf{M}^\star \in \arg\min_{\mathbf{M}:|\mathbf{M}|\leq B_{\mathrm{mem}}} \left[D_q(\mathbf{C}, \mathbf{M}) + \lambda_{\mathrm{RD}} c_{\mathrm{mem}}(\mathbf{M})\right]. \tag{45}$$

WATCH does not solve Eq. 45 explicitly. Instead, its learned memory policy amortizes the corresponding task–compression trade-off from downstream reinforcement-learning feedback.

The distortion also controls the change of any bounded answer-level utility.

Lemma 3 (Answer-utility preservation)**.** *Let $f(y) \in [0, 1]$ be any bounded measurable score of the generated answer. Then*

$$|\mathbb{E}_{P_{\mathbf{C}}}[f(y)] - \mathbb{E}_{P_{\mathbf{M}}}[f(y)]| \leq \sqrt{\frac{1}{2} D_q(\mathbf{C}, \mathbf{M})}. \tag{46}$$

Proof. For any measurable f taking values in $[0, 1]$,

$$|\mathbb{E}_{P_{\mathbf{C}}}[f] - \mathbb{E}_{P_{\mathbf{M}}}[f]| \leq D_{\mathrm{TV}}(P_{\mathbf{C}}, P_{\mathbf{M}}).$$

Pinsker's inequality gives

$$D_{\mathrm{TV}}(P_{\mathbf{C}}, P_{\mathbf{M}}) \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}(P_{\mathbf{C}} \| P_{\mathbf{M}})}.$$

Substituting Eq. 44 completes the proof. □

Lemma 3 formalizes the quantity targeted by task-aware compression: when the compressed memory induces an answer distribution close to that of the candidate evidence, every bounded answer-level utility changes by a correspondingly bounded amount. This motivates conditioning retention on query-dependent evidence, complementary coverage, and redundancy rather than on a fixed token-importance rule.

D Reinforcement-Learning and Training Details

This appendix expands the learning procedure in Section 3.3. We first formalize the potential-based credit construction and its continuation-value estimator, then describe the memory-aware demonstration warm-start, heterogeneous policy update, and budget-conditioned training schedule.

D.1 Potential-Based Credit and Continuation Value

WATCH contains two internal transitions for every successful continuation:

$$\bar{\zeta}_k^M \xrightarrow{\text{acquire}} \bar{\zeta}_{k+1}^C \xrightarrow{\text{retain}} \bar{\zeta}_{k+1}^M.$$

Here

$$\bar{\zeta}_k^M = (\mathrm{M}, q, \mathbf{M}_k, \mathbf{L}_k, \boldsymbol{\beta}_k)$$

is the pre-acquisition shaping state, whereas

$$\bar{\zeta}_{k+1}^C = \big(\mathrm{C}, q, \mathbf{C}_{k+1}, \mathbf{L}_{k+1}^C, \boldsymbol{\beta}_k^+\big)$$

is the candidate shaping state after the k-th view has been acquired but before memory control. The symbols M and C are stage indicators. These shaping states deliberately exclude learned policy activations such as $\mathbf{h}_k$. We introduce an absorbing terminal state † with $\Phi_\xi(\dagger) = 0$.

For compactness, write

$$\Phi_k^M = \Phi_\xi(\bar{\zeta}_k^M), \qquad \Phi_{k+1}^C = \Phi_\xi(\bar{\zeta}_{k+1}^C).$$

The shaping rewards are

$$F_k^{\text{acq}} = \Phi_{k+1}^C - \Phi_k^M, \qquad F_k^{\text{mem}} = \Phi_{k+1}^M - \Phi_{k+1}^C, \qquad F^{\text{halt}} = -\Phi_K^M. \tag{47}$$

Proof of Proposition 2. For a trajectory with K adaptive acquisitions,

$$\begin{aligned}\sum_{k=0}^{K-1} (F_k^{\text{acq}} + F_k^{\text{mem}}) &= \sum_{k=0}^{K-1} \big(\Phi_{k+1}^M - \Phi_k^M\big) \\ &= \Phi_K^M - \Phi_0^M.\end{aligned}$$

Adding the terminal shaping reward gives

$$\sum_{k=0}^{K-1} (F_k^{\text{acq}} + F_k^{\text{mem}}) + F^{\text{halt}} = -\Phi_0^M.$$

For a fixed input and prescribed budget, the query q, the deterministic preview banks $(\mathbf{M}_0, \mathbf{L}_0)$, and the initial resource descriptor $\boldsymbol{\beta}_0$ are identical across policies. Hence $\bar{\zeta}_0^M$ is policy independent. Moreover, the potential parameters ξ are disjoint from the WATCH policy parameters θ and are held fixed during the policy comparison. Therefore adding ηF shifts every trajectory return for the same context by the same constant $-\eta\Phi_\xi(\bar{\zeta}_0^M)$ and cannot change the ordering of policies or the set of optimal conditional policies. □

The result concerns the underlying reward transformation. The clipped finite-sample policy update used in practice remains an optimization approximation and is not itself covered by the policy-invariance statement. When Φ_ξ is updated during training, Proposition 2 applies conditionally to each controller update for which ξ is held fixed (Ng et al., 1999).

Continuation-return targets. For rollout g, let K_g denote its number of acquisitions and abbreviate

$$Q_g = Q(x, y^{(g)}), \qquad C_{g,t}^{\text{obs}} = c_{\text{obs}}(v_{g,t}).$$

The realized continuation utility from a pre-acquisition shaping state $\bar{\zeta}_{g,k}^M$ is

$$G_{g,k}^M = Q_g - \lambda_{\text{mem}} c_{\text{mem}}(\mathbf{M}_{K_g}^{(g)}) - \lambda_{\text{obs}} \sum_{t=k}^{K_g-1} C_{g,t}^{\text{obs}}. \tag{48}$$

After view $v_{g,k}$ has already been executed, its cost is sunk. The continuation utility from the corresponding candidate shaping state is therefore

$$G^C_{g,k+1} = Q_g - \lambda_{\text{mem}} c_{\text{mem}}(\mathbf{M}^{(g)}_{K_g}) - \lambda_{\text{obs}} \sum_{t=k+1}^{K_g-1} C^{\text{obs}}_{g,t}. \tag{49}$$

For a terminal memory state, the observation-cost sum is empty.

These targets deliberately exclude computation incurred before the shaping state at which the value is evaluated. Consequently, Φ_ξ approximates a continuation value consistent with the Bellman interpretation in Section 3.1, rather than a regression to the full trajectory reward from time zero.

Potential estimator. For a generic shaping state

$$\bar{\zeta} = \left(\sigma, q, \mathbf{X}, \mathbf{L}^X, \boldsymbol{\beta}\right),$$

where $\mathbf{X} = \{\mathbf{x}_i\}$ is either a persistent feature bank or a candidate feature bank and $\mathbf{L}^X = \{\boldsymbol{\ell}_i\}$ is its aligned controller-side metadata bank, we use

$$\Phi_\xi(\bar{\zeta}) = f_\xi\left(\left[E^\Phi_q(q); \text{Pool}_\Phi\left(\{[P_\Phi(\mathbf{x}_i); E^\Phi_\ell(\boldsymbol{\ell}_i)]\}_i\right); E^\Phi_\beta(\boldsymbol{\beta}); E^\Phi_\sigma(\sigma)\right]\right). \tag{50}$$

Here f_ξ, E^Φ_q, P_Φ, E^Φ_ℓ, E^Φ_β, E^Φ_σ, and Pool_Φ are lightweight potential-specific modules whose trainable parameters are included in ξ. None shares trainable parameters with the WATCH policy π_θ.

Let $\mathcal{E}^M_g$ and $\mathcal{E}^C_g$ denote the memory and candidate shaping states visited by rollout g. The estimator is trained with

$$\mathcal{L}_\Phi = \frac{1}{N_\Phi} \sum_g \left[\sum_{k \in \mathcal{E}^M_g} \text{Huber}\left(\Phi_\xi(\bar{\zeta}^M_{g,k}) - \text{sg}[G^M_{g,k}]\right) + \sum_{k \in \mathcal{E}^C_g} \text{Huber}\left(\Phi_\xi(\bar{\zeta}^C_{g,k}) - \text{sg}[G^C_{g,k}]\right) \right], \tag{51}$$

where N_Φ is the number of supervised shaping states, $\text{sg}[\cdot]$ denotes stop-gradient, and $\text{Huber}(\cdot)$ is the Huber regression loss. The potential is updated between controller updates and held fixed while optimizing the policy on the corresponding rollout batch. It is discarded at inference time.

For an action associated with pre-action shaping state $\bar{\zeta}_{g,u}$, define

$$\widetilde{G}_{g,u} = G_{g,u} - \eta \Phi_\xi(\bar{\zeta}_{g,u}). \tag{52}$$

This is a state-dependent potential baseline, not an action-dependent one.

For decision type $d \in \{\text{halt}, \text{view}, \text{mem}\}$ and step k, let $\mathcal{G}^d_k$ be the subset of group rollouts that reach that decision stage. We use

$$A^d_{g,k} = \frac{\widetilde{G}^d_{g,k} - \mu^d_k}{\sigma^d_k + \epsilon}, \qquad g \in \mathcal{G}^d_k, \tag{53}$$

where μ^d_k and σ^d_k are computed over $\mathcal{G}^d_k$. If fewer than two rollouts reach a stage or its empirical variance is numerically zero, we fall back to centered batch normalization over the same decision type.

The halt and view decisions at step k use the pre-acquisition return $G^M_{g,k}$ and shaping state $\bar{\zeta}^M_{g,k}$; the view advantage is normalized only over trajectories that continue. All memory-policy rounds generated from $\mathbf{C}_{k+1}$ share the memory-stage scalar advantage derived from $G^C_{g,k+1}$ and $\bar{\zeta}^C_{g,k+1}$, while retaining distinct autoregressive policy ratios.

D.2 Offline Demonstration Warm-Start

The structured view space and variable-length anchor policy make purely random initial exploration inefficient. We therefore construct a modest offline demonstration set $\mathcal{D}_{\text{demo}}$ across the target range of $(B_{\text{obs}}, B_{\text{mem}})$ and use it only to initialize the controller. Demonstrations follow the same causal ordering as WATCH: a memory teacher first defines how candidate evidence is compressed under the prescribed capacity, and an acquisition teacher evaluates candidate views through this memory bottleneck.

Memory teacher. For demonstration construction only, relevance, novelty, and redundancy are evaluated in a fixed teacher embedding space that is independent of the learned controller projection P_c. Let

$$T_v^{\text{teach}} : \mathbb{R}^{d_v} \to \mathbb{R}^{d_t}, \qquad T_q^{\text{teach}} : q \mapsto \mathbb{R}^{d_t}$$

denote frozen visual and textual maps into a shared teacher space. These maps contain no trainable WATCH parameters and are fixed before demonstration construction; their concrete instantiation is specified together with the backbone configuration.

For candidate i, define

$$\bar{\mathbf{c}}_i^{\text{teach}} = T_v^{\text{teach}}(\mathbf{c}_i), \qquad \bar{\mathbf{q}}^{\text{teach}} = T_q^{\text{teach}}(q),$$

and its teacher relevance as

$$r_i^{\text{teach}} = \cos\left(\bar{\mathbf{c}}_i^{\text{teach}}, \bar{\mathbf{q}}^{\text{teach}}\right). \tag{54}$$

For newly acquired tokens, we analogously define

$$\nu_i^{\text{teach}} = \frac{1}{2}\left[1 - \max_{\mathbf{m}\in\mathbf{M}} \cos\left(\bar{\mathbf{c}}_i^{\text{teach}}, T_v^{\text{teach}}(\mathbf{m})\right)\right],$$

with $\nu_i^{\text{teach}} = 1$ when $\mathbf{M} = \varnothing$ and $\nu_i^{\text{teach}} = 0$ for inherited tokens.

To encourage complementary anchors, let d_{ij}^{geo} be the geometric discrepancy defined in Eq. 38, and define

$$c_i^{\text{geo}}(\mathcal{S}) = \begin{cases} 0, & \mathcal{S} = \varnothing, \\ \min\limits_{j\in\mathcal{S}} \min\left\{\dfrac{d_{ij}^{\text{geo}}}{\tau_{\text{geo}}}, 1\right\}, & \text{otherwise.} \end{cases}$$

Given the anchors already selected, the teacher scores candidate i by

$$S_{\text{mem}}^{\text{teach}}(i \mid \mathcal{S}) = \alpha_q r_i^{\text{teach}} + \alpha_\nu \nu_i^{\text{teach}} + \alpha_c c_i^{\text{geo}}(\mathcal{S}) - \alpha_r \max_{j\in\mathcal{S}} \cos\left(\bar{\mathbf{c}}_i^{\text{teach}}, \bar{\mathbf{c}}_j^{\text{teach}}\right), \tag{55}$$

where the final redundancy term is zero when $\mathcal{S} = \varnothing$.

The first anchor is always the highest-scoring candidate. Thereafter, the teacher repeatedly adds the highest-scoring unselected candidate while its score exceeds a validation-selected threshold $\tau_{\text{mem}}^{\text{teach}}$, terminating with a STOP action otherwise or when the capacity

$$B^a = \min\{B_{\text{mem}}, N^c\}$$

is reached. This produces an ordered demonstration sequence compatible with the autoregressive memory policy in Appendix C.4.

Given the resulting anchor set, the teacher applies the deterministic geometry-aware merge/prune rule in Appendix C.4, with semantic similarities evaluated in the same fixed teacher embedding space. We denote the resulting compression operator by

$$(\mathbf{M}_{\text{teach}}, \mathbf{L}_{\text{teach}}) = \mathcal{K}_{\text{teach}}\left(\mathbf{C}, \mathbf{L}^C; q, B_{\text{mem}}\right).$$

This rule is used only to construct demonstrations and does not constrain the learned memory policy after warm-start.

Acquisition teacher. At each demonstration state, a budget-feasible greedy teacher evaluates a finite proposal set containing global views, multi-scale local crops, and admissible temporal windows. We denote its offline quality evaluator by $\widehat{Q}_{\text{teach}}(q, \mathbf{X})$. When reference answers are available, the default implementation uses normalized reference-answer log-likelihood under the frozen backbone,

$$\widehat{Q}_{\text{teach}}(q, \mathbf{X}; y^\star) = \frac{1}{|y^\star|} \sum_{j=1}^{|y^\star|} \log \pi_\phi\left(y_j^\star \mid y_{<j}^\star, q, \mathbf{X}\right), \tag{56}$$

while task-specific automatic evaluators may be substituted when appropriate. This quantity is used only during offline demonstration construction.

Define the teacher-side halt utility

$$\widehat{H}_{\text{teach}}(q, \mathbf{M}) = \widehat{Q}_{\text{teach}}(q, \mathbf{M}) - \lambda_{\text{mem}} c_{\text{mem}}(\mathbf{M}). \tag{57}$$

For a feasible candidate view v, let

$$\widetilde{\mathbf{Z}}(v) = E_v\big(\mathcal{O}(\mathcal{V}, v)\big), \qquad \widetilde{\mathbf{L}}(v) = \text{Meta}\left(v, \widetilde{\mathbf{Z}}(v)\right),$$

and form

$$\mathbf{C}(v) = [\mathbf{M}; \widetilde{\mathbf{Z}}(v)], \qquad \mathbf{L}^C(v) = [\mathbf{L}; \widetilde{\mathbf{L}}(v)].$$

The teacher then compresses the proposal under the actual memory capacity,

$$\left(\mathbf{M}'_{\text{teach}}(v), \mathbf{L}'_{\text{teach}}(v)\right) = \mathcal{K}_{\text{teach}}\left(\mathbf{C}(v), \mathbf{L}^C(v); q, B_{\text{mem}}\right), \tag{58}$$

and scores the view by the one-step utility improvement

$$S_{\text{view}}(v) = \widehat{H}_{\text{teach}}\left(q, \mathbf{M}'_{\text{teach}}(v)\right) - \widehat{H}_{\text{teach}}(q, \mathbf{M}) - \lambda_{\text{obs}} c_{\text{obs}}(v). \tag{59}$$

Thus a view receives credit only for evidence that remains useful after the prescribed memory bottleneck. The teacher executes the highest-scoring feasible view when its score exceeds a validation-selected margin $\tau^{\text{teach}}_{\text{view}}$ and otherwise emits the halt action. After acquisition, the corresponding teacher memory becomes the state used to construct the next demonstration decision.

Behavior cloning. For each continued demonstration step, let $\mathbf{a}^{\text{anc},\star}_{k+1} = \left(a^{\text{anc},\star}_{k+1,1}, \ldots, a^{\text{anc},\star}_{k+1,R^\star_{k+1}}\right)$ denote the ordered memory-policy decisions produced by the memory teacher. If the teacher terminates before reaching capacity, the final decision is STOP; otherwise the sequence ends at the capacity limit.

Let $\psi^{\text{mem},\star}_{k+1,r}$ denote the teacher-forced policy context at round r, containing the contextualized candidate bank, post-acquisition decision context, and previously selected demonstration anchors. Behavior cloning minimizes

$$\begin{aligned}\mathcal{L}_{\text{BC}} = -\mathbb{E}_{\tau^\star \sim \mathcal{D}_{\text{demo}}}\Bigg[&\sum_k \log \pi^{\text{halt}}_\theta\left(a^{\text{halt},\star}_k \mid \mathbf{d}_k\right) + \sum_{k: a^{\text{halt},\star}_k = 0} \log \pi^{\text{view}}_\theta\left(v^\star_k \mid \mathbf{d}_k\right) \\ &+ \lambda^{\text{mem}}_{\text{BC}} \sum_{k: a^{\text{halt},\star}_k = 0} \frac{1}{R^\star_{k+1}} \sum_{r=1}^{R^\star_{k+1}} \log \pi^{\text{mem}}_\theta\left(a^{\text{anc},\star}_{k+1,r} \mid \psi^{\text{mem},\star}_{k+1,r}\right)\Bigg].\end{aligned} \tag{60}$$

The demonstrations provide only a valid and economical initialization; subsequent reinforcement learning is free to depart from the teacher.

D.3 HETEROGENEOUS POLICY UPDATE

WATCH contains heterogeneous stochastic actions, so the implementation uses action-specific likelihood ratios rather than treating an entire perception step as one monolithic action.

Halting and view allocation. For the halt decision,

$$\rho^{\text{halt}}_{g,k} = \frac{\pi^{\text{halt}}_\theta(a^{\text{halt}}_{g,k} \mid \mathbf{d}_{g,k})}{\pi^{\text{halt}}_{\theta_{\text{old}}}(a^{\text{halt}}_{g,k} \mid \mathbf{d}_{g,k})}. \tag{61}$$

For the canonical view $v_k = (g_k, I_k, R_k, \rho_k)$,

$$\begin{aligned}\log \pi^{\text{view}}_\theta(v_k \mid \mathbf{d}_k) = \ &\log \pi^g_\theta(g_k \mid \mathbf{d}_k) + \log \pi^I_\theta(I_k \mid g_k, \mathbf{d}_k) \\ &+ \mathbf{1}[g_k = \text{local}] \log \pi^R_\theta(R_k \mid g_k, I_k, \mathbf{d}_k, b_k) + \log \pi^\rho_\theta(\rho_k \mid g_k, I_k, R_k, \mathbf{d}_k, b_k).\end{aligned} \tag{62}$$

Discrete probabilities are normalized only over feasible choices. For the continuous components, the support transformations in Appendix C.2 depend on the realized state and preceding action prefix but not on the optimized policy parameters. Their Jacobian factors therefore cancel in the new-to-old importance ratio, so the ratio may equivalently be computed from the corresponding latent Beta densities.

The view ratio is

$$\rho^{\text{view}}_{g,k} = \exp\left(\log \pi^{\text{view}}_\theta(v_{g,k} \mid \mathbf{d}_{g,k}) - \log \pi^{\text{view}}_{\theta_{\text{old}}}(v_{g,k} \mid \mathbf{d}_{g,k})\right).$$

Autoregressive memory control. Let $\psi^{\mathrm{mem}}_{g,k,r} = \left(\{\mathbf{u}_{g,k+1,i}\}_{i=1}^{N^c_{g,k+1}}, \mathbf{d}^+_{g,k}, \mathcal{S}^{(r-1)}_{g,k+1}\right)$ denote the policy context before anchor-selection round r. The realized action $a^{\mathrm{anc}}_{g,k+1,r}$ is either an unselected candidate index or STOP, with the support defined in Appendix C.4. Its likelihood ratio is

$$\rho^{\mathrm{mem}}_{g,k,r} = \frac{\pi^{\mathrm{mem}}_{\theta}\left(a^{\mathrm{anc}}_{g,k+1,r} \mid \psi^{\mathrm{mem}}_{g,k,r}\right)}{\pi^{\mathrm{mem}}_{\theta_{\mathrm{old}}}\left(a^{\mathrm{anc}}_{g,k+1,r} \mid \psi^{\mathrm{mem}}_{g,k,r}\right)}. \tag{63}$$

Equivalently, if $R^{\mathrm{mem}}_{g,k}$ memory-policy rounds are realized at the k-th memory update, the sequence log-probability factorizes as

$$\log \pi^{\mathrm{mem}}_{\theta}(a^{\mathrm{mem}}_{g,k}) = \sum_{r=1}^{R^{\mathrm{mem}}_{g,k}} \log \pi^{\mathrm{mem}}_{\theta}\left(a^{\mathrm{anc}}_{g,k+1,r} \mid \psi^{\mathrm{mem}}_{g,k,r}\right). \tag{64}$$

We optimize the factors separately rather than forming a single product ratio over the entire variable-length sequence. All rounds share the memory-stage advantage $A^{\mathrm{mem}}_{g,k}$, while the autoregressive contexts and likelihood ratios remain round specific.

Define

$$\mathrm{ClipPG}(\rho, A) = \min\left(\rho A,\ \mathrm{clip}(\rho, 1-\varepsilon, 1+\varepsilon)A\right).$$

The full policy loss is

$$\begin{aligned}\mathcal{L}_{\mathrm{policy}} = -\,\mathbb{E}\Bigg[& \mathrm{ClipPG}\left(\rho^{\mathrm{halt}}, A^{\mathrm{halt}}\right) \\ & + \mathbf{1}[a^{\mathrm{halt}}=0]\,\mathrm{ClipPG}\left(\rho^{\mathrm{view}}, A^{\mathrm{view}}\right) \\ & + \mathbf{1}[a^{\mathrm{halt}}=0]\frac{\lambda^{\mathrm{RL}}_{\mathrm{mem}}}{R^{\mathrm{mem}}}\sum_{r=1}^{R^{\mathrm{mem}}}\mathrm{ClipPG}\left(\rho^{\mathrm{mem}}_{r}, A^{\mathrm{mem}}\right)\Bigg] \\ & + \beta_{\mathrm{KL}}\widehat{D}_{\mathrm{KL}}(\pi_\theta\|\pi_{\mathrm{ref}}) - \beta_{\mathrm{ent}}\widehat{\mathcal{H}}(\pi_\theta),\end{aligned} \tag{65}$$

where π_{ref} is the frozen warm-start policy. $\widehat{D}_{\mathrm{KL}}$ and $\widehat{\mathcal{H}}$ denote empirical averages over the same heterogeneous policy factors and are used only for stabilization.

Averaging the memory term over realized memory-policy rounds prevents longer memory-action sequences from receiving disproportionate optimization weight. The environment transition depends only on the completed anchor set and the deterministic geometry-aware merge/prune rule; all stochastic parameter-dependent decisions are therefore explicitly represented in the policy likelihood.

D.4 BUDGET-CONDITIONED TRAINING SCHEDULE

Training proceeds in three phases.

Phase I: demonstration warm-start. We first optimize Eq. 60 on demonstrations sampled across the supported range of $(B_{\mathrm{obs}}, B_{\mathrm{mem}})$. The potential estimator is initialized from the continuation-return targets induced by these trajectories. After warm-start, we freeze a copy of the resulting controller as the reference policy π_{ref} used by the KL regularizer in Eq. 65.

Phase II: budget curriculum. On-policy training begins with comparatively permissive observation and memory budgets, where useful evidence can be acquired reliably. The lower endpoints of the sampled budget distribution are then gradually annealed toward the target deployment range. Only the budget distribution changes; the view and memory feasibility rules remain identical to those used at deployment.

Phase III: full-range on-policy optimization. After the curriculum reaches the target range, budgets are sampled from the full deployment distribution. Each policy iteration begins by snapshotting

the current controller, $\theta_{\text{old}} \leftarrow \theta$, and collecting G trajectories for each sampled $(x, B_{\text{obs}}, B_{\text{mem}})$ under $\pi_{\theta_{\text{old}}}$.

The controller is held fixed while the potential parameters ξ are updated from the resulting continuation-return targets using Eq. 51. We then freeze ξ for the corresponding policy update, construct group-relative advantages with Eq. 53, and optimize Eq. 65 for a small number of mini-batch epochs using importance ratios against $\pi_{\theta_{\text{old}}}$ and KL regularization toward the fixed π_{ref}. A new snapshot is created only after the current policy-update cycle is complete.

The visual front-end and language backbone remain frozen unless explicitly stated otherwise. This procedure yields a single budget-conditioned WATCH controller rather than training separate policies for individual observation or memory operating points.

E Inference Protocol and Complexity Analysis

This appendix expands the deterministic deployment procedure summarized in Algorithm 1. We first specify how the stochastic controller is converted into a hard-budget policy, then account for visual and controller computation, and finally describe system-level latency and video-budget comparisons.

E.1 Deterministic Budgeted Execution

Inference begins from the same policy-independent preview as training:

$$(\mathbf{M}_0, \mathbf{L}_0) = \mathcal{P}_0\big(E_v(\mathcal{O}_0(\mathcal{V})); B_{\text{mem}}\big), \qquad b_0 = B_{\text{obs}} - c_{\text{pre}}.$$

Budgets below c_{pre} are outside the supported operating range.

At decision step k, WATCH halts if $p_k^{\text{halt}} \geq \tau_{\text{halt}}$ or if $\mathcal{F}_{\text{view}}(z_k, b_k) = \varnothing$. The threshold τ_{halt} is selected on held-out validation data.

Otherwise, the view policy is executed autoregressively. Among feasible discrete choices, WATCH selects the maximum-probability g_k and I_k. For a latent $\upsilon \sim \text{Beta}(\alpha, \beta)$, stochastic sampling is replaced by its mean

$$\bar{\upsilon} = \frac{\alpha}{\alpha + \beta}.$$

Applying this mean through the budget-feasible transformations in Appendix C.2 deterministically produces R_k and ρ_k without any post-hoc projection.

After acquisition, the memory controller first contextualizes the candidate bank as in Appendix C.3. Let $B_{k+1}^a = \min\{B_{\text{mem}}, N_{k+1}^c\}$. Anchor decoding then follows the same autoregressive support as training. At the first round, STOP is masked and WATCH selects the highest-probability candidate. At each subsequent round, it greedily chooses the highest-probability action among the remaining candidates and STOP. Decoding terminates when STOP is selected or when B_{k+1}^a anchors have been retained. Consequently,

$$1 \leq |\mathcal{S}_{k+1}| \leq B_{k+1}^a \leq B_{\text{mem}}$$

without a separate threshold or feasibility repair.

Non-anchor candidates are then processed by the geometry-aware redundancy rule in Appendix C.4: a candidate is merged only with a semantically redundant and geometrically compatible anchor, and is otherwise pruned. The resulting aligned banks therefore satisfy

$$1 \leq |\mathbf{M}_{k+1}| = |\mathbf{L}_{k+1}| \leq B_{\text{mem}}$$

at every adaptive step.

The language backbone is never called inside this perception loop. The aligned metadata bank $\mathbf{L}_k$ remains controller-side throughout inference and does not contribute visual tokens to the backbone. Once halting is triggered or no further feasible observation remains, the final response is generated once from $\mathbf{M}_K$:

$$y \sim \pi_\phi(\cdot \mid q, \mathbf{M}_K).$$

Figure 4 shows an illustrative WATCH rollout, where successive acquisitions improve evidence quality while the memory size remains bounded.

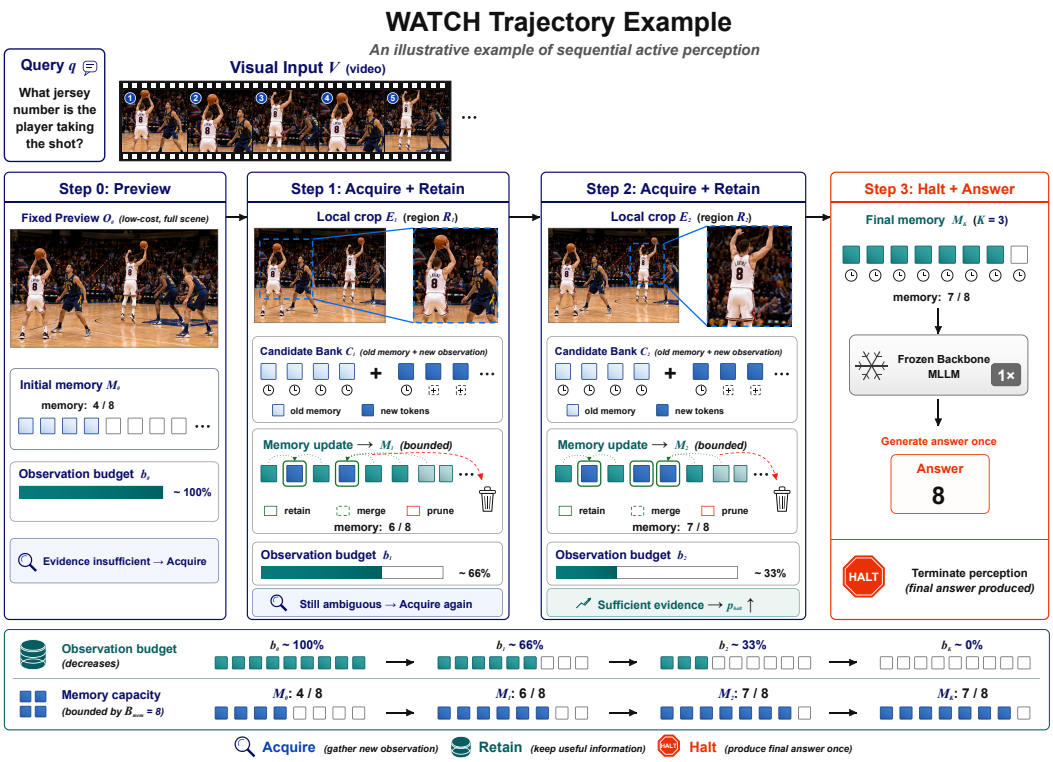


Figure 4: An illustrative WATCH trajectory. The example demonstrates how sequential Acquire–Retain decisions refine visual evidence until the halting criterion is satisfied and the frozen language backbone is invoked once.

E.2 Resource Accounting and Computational Complexity

WATCH controls computation both before and after visual encoding, so a terminal token-retention ratio alone is insufficient to characterize efficiency.

Let $N_{\text{pre}}^{\text{enc}}$ denote the number of tokens produced by the visual front-end for the fixed preview *before* the deterministic reducer $\mathcal{P}_0$. The total number of visual tokens encoded during one inference is

$$N_{\text{enc}} = N_{\text{pre}}^{\text{enc}} + \sum_{k=0}^{K-1} |\widetilde{\mathbf{Z}}_{k+1}|. \tag{66}$$

This quantity measures input-side visual processing, including repeated inspection.

The cumulative candidate volume handled by the learned memory controller is

$$N_{\text{ctrl}} = \sum_{k=0}^{K-1} N_{k+1}^{c}, \tag{67}$$

and the total number of autoregressive memory-policy rounds is

$$R_{\text{mem}} = \sum_{k=0}^{K-1} R_{k}^{\text{mem}}, \tag{68}$$

where R_k^{mem} includes a terminating STOP action when the k-th memory update stops before reaching capacity. The expensive backbone receives only

$$N_{\text{backbone}} = |\mathbf{M}_K| \le B_{\text{mem}}. \tag{69}$$

Persistent memory is bounded independently of trajectory length. If every adaptive observation produces at most $N_{\text{obs}}^{\max}$ tokens, then

$$N_{k+1}^{c} = |\mathbf{M}_k| + |\widetilde{\mathbf{Z}}_{k+1}| \le B_{\text{mem}} + N_{\text{obs}}^{\max}.$$

Moreover,

$$R_k^{\mathrm{mem}} \le B_{k+1}^a \le B_{\mathrm{mem}}.$$

Thus neither persistent memory, the candidate bank at a single step, nor the number of sequential memory decisions at that step can grow without bound.

Let $\mathcal{C}_v(n)$ denote the cost of visual encoding that produces n tokens, and let $\mathcal{C}_{\mathrm{ctrl}}(N, R, J)$ denote the lightweight controller cost for a candidate bank of size N, R realized memory-policy rounds, and J retained anchors. Total inference computation can be decomposed as

$$\begin{aligned}\mathcal{C}_{\mathrm{WATCH}} &= \mathcal{C}_{\mathrm{pre}} + \sum_{k=0}^{K-1} \mathcal{C}_v\big(|\widetilde{\mathbf{Z}}_{k+1}|\big) \\ &+ \sum_{k=0}^{K-1} \mathcal{C}_{\mathrm{ctrl}}\left(N_{k+1}^c, R_k^{\mathrm{mem}}, J_{k+1}\right) + \mathcal{C}_{\mathrm{backbone}}(N_q + N_{\mathrm{backbone}}, L_y),\end{aligned} \tag{70}$$

where $\mathcal{C}_{\mathrm{pre}}$ includes preview encoding and fixed reduction, N_q is the textual context length, and L_y is the generated response length.

We next separate the dominant components of $\mathcal{C}_{\mathrm{ctrl}}$. If the candidate contextualizer contains L_m shallow self-attention layers of width d_c, its token-interaction cost is

$$\mathcal{O}\left(L_m (N_{k+1}^c)^2 d_c\right).$$

Let d_a denote the width of the sequential anchor selector. At memory round r, at most $N_{k+1}^c - r + 1$ unselected candidates are scored. Hence the anchor-decoding cost satisfies

$$\begin{aligned}\mathcal{C}_{\mathrm{anchor}} &= \mathcal{O}\left(d_a \sum_{r=1}^{R_k^{\mathrm{mem}}} (N_{k+1}^c - r + 1)\right) \\ &\subseteq \mathcal{O}\left(R_k^{\mathrm{mem}} N_{k+1}^c d_a\right) \\ &\subseteq \mathcal{O}\left(B_{k+1}^a N_{k+1}^c d_a\right).\end{aligned} \tag{71}$$

When feature-space merging is enabled, comparing non-anchor candidates with at most J_{k+1} retained anchors contributes

$$\mathcal{O}\left(J_{k+1} N_{k+1}^c d_c\right)$$

for semantic redundancy resolution. Consequently, the dominant memory-controller scaling at one adaptive step is

$$\mathcal{C}_{\mathrm{ctrl}} = \mathcal{O}\Big(L_m (N_{k+1}^c)^2 d_c + R_k^{\mathrm{mem}} N_{k+1}^c d_a + J_{k+1} N_{k+1}^c d_c\Big), \tag{72}$$

up to lower-order recurrent-state and policy-head operations.

Using

$$N_{k+1}^c \le B_{\mathrm{mem}} + N_{\mathrm{obs}}^{\max}, \qquad R_k^{\mathrm{mem}}, J_{k+1} \le B_{\mathrm{mem}},$$

gives the worst-case bound

$$\mathcal{O}\left(L_m (B_{\mathrm{mem}} + N_{\mathrm{obs}}^{\max})^2 d_c + B_{\mathrm{mem}} (B_{\mathrm{mem}} + N_{\mathrm{obs}}^{\max})(d_a + d_c)\right).$$

For an L_ϕ-layer Transformer backbone with hidden dimension d_ϕ, the attention-dominated prefill contribution scales approximately as

$$\mathcal{C}_{\mathrm{prefill}} = \mathcal{O}\left(L_\phi (N_q + N_{\mathrm{backbone}})^2 d_\phi\right), \tag{73}$$

while KV-cached autoregressive decoding contributes approximately

$$\mathcal{O}\left(L_\phi d_\phi \sum_{j=1}^{L_y} (N_q + N_{\mathrm{backbone}} + j)\right)$$

to attention computation. These expressions describe dominant scaling rather than exact FLOPs; projection, MLP, visual-front-end, controller, and system overheads remain implementation dependent.

If exhaustive visual processing would deliver N_{full} tokens to the backbone, the terminal context ratio

$$r_{\text{ctx}} = \frac{N_{\text{backbone}}}{N_{\text{full}}}$$

characterizes backbone-side visual-context reduction. It should not be interpreted directly as wall-clock speedup because WATCH also incurs adaptive visual encoding and lightweight sequential control.

F Experimental Details

This section specifies the complete experimental protocol used throughout Section 4. Unless explicitly marked as a published-model reference, all controlled comparisons use the same raw inputs, frozen backbone, terminal task prompts, answer decoding, and evaluation pipeline. Method-specific internal search or selection procedures follow their respective original implementations.

F.1 Backbones and Visual Interface

Primary backbone. Our controlled experiments use Qwen2.5-VL-7B-Instruct (Bai et al., 2025b). Both the visual front-end E_v and the language backbone π_ϕ are frozen throughout WATCH training. We define E_v to include the native vision encoder and multimodal projector up to the backbone-compatible visual-token interface used by WATCH. Consequently, acquired and merged features can be passed to the language backbone without an additional learned visual adapter. The language backbone is not invoked during the adaptive perception loop and is called once on the terminal memory M_K.

Terminal memory packing. Because the terminal memory may contain tokens acquired from different views, we preserve their original spatial–temporal locations when interfacing with the language backbone. For each retained token with metadata $\ell_i = (t_i, x_i, y_i, \rho_i)$, we map (x_i, y_i) to a backbone-native full-frame reference grid (H_ϕ, W_ϕ) determined by the native visual preprocessing of the corresponding input, and use

$$p_i^h = \text{round}(y_i(H_\phi - 1)), \qquad p_i^w = \text{round}(x_i(W_\phi - 1)).$$

For images, $p_i^t = 0$; for videos, p_i^t is obtained from the token's original timestamp using the backbone's native temporal position convention. Retained features are ordered temporally and then spatially and inserted into exactly $|M_K|$ visual slots with the resulting backbone-native position indices. A merged token inherits the position of its retained anchor, consistent with the metadata update in Appendix C.4. The metadata themselves are never embedded as additional visual tokens. For Qwen3-VL, we use the same procedure with its native interleaved M-RoPE interface.

The query embedding used by the lightweight controller is obtained by mean-pooling the frozen Qwen token embeddings of the query and projecting the result to 512 dimensions with a trainable controller-side linear layer. This operation does not execute the language Transformer. All backbone inference uses BF16 and FlashAttention-2 (Dao, 2024).

Second-backbone evaluation. To test whether the gains depend on a particular visual backbone, we repeat the main experiments with Qwen3-VL-8B-Instruct (Bai et al., 2025a). Qwen3-VL remains frozen; we retrain only WATCH and the backbone-specific input projections using the same data mixture, budget distribution, and optimization schedule. Controller hidden dimensions and all resource definitions are unchanged. Thus this experiment evaluates architectural transfer rather than zero-shot transfer of controller weights between incompatible visual feature spaces.

F.2 Benchmarks and Task Metrics

Image tasks. We use V*Bench (Wu & Xie, 2024) to measure targeted visual search under fine-grained and cluttered imagery, and evaluate both the 4K and 8K variants of HR-Bench (Wang et al., 2025a) to stress high-resolution local perception. DocVQA (Mathew et al., 2021) evaluates document-level reading and localized textual evidence, while RealWorldQA (xAI, 2024) and MMMU (Yue et al., 2024) test whether active perception and bounded memory preserve general

multimodal reasoning. We use the official evaluation split when labels are public and the official evaluation server otherwise.

Video tasks. Video-MME (Fu et al., 2025) evaluates short-, medium-, and long-duration video understanding. Our primary efficiency comparison uses the visual-only setting, with video frames but without subtitles or audio, so that changes in performance can be attributed to visual evidence allocation. LongVideoBench (Wu et al., 2024) is evaluated using its official interleaved frame–subtitle input. The same subtitle stream is provided to every method and is not charged to B_{obs}; only visual observations consume the visual acquisition budget. EgoSchema (Mangalam et al., 2023) probes long-horizon egocentric understanding, and NExT-QA (Xiao et al., 2021) emphasizes causal and temporal reasoning.

Quality metrics. We follow each benchmark's official evaluation protocol and report all scores on a $[0, 100]$ scale. DocVQA is evaluated with ANLS, while HR-Bench uses its official cyclic-permutation evaluation over the answer choices. Other multiple-choice tasks are scored by answer correctness after the benchmark-specific deterministic parser, and short free-form answers use the corresponding normalized matching rule. For MMMU, we use the released official parsing and evaluation code. For RealWorldQA, we deterministically extract option letters when applicable and otherwise normalize the released short-answer strings by case, surrounding whitespace, and punctuation; no external LLM judge is used. Controller-training rewards are defined on the training sources in Appendix F.3 and scaled to $Q(x, y) \in [0, 1]$ as specified in Appendix F.6.

For system-efficiency measurements on HR-Bench, we time one canonical answer ordering per underlying question; cyclic permutations are used only for computing the official task score. Thus benchmark-side repeated evaluation does not artificially multiply the per-request resource measurements reported in Appendix H.

F.3 Controller Training Data

Training mixture. Controller training is intentionally separated from the headline evaluation suites. Before constructing the controller-validation split, we sample without replacement a 160K-example image pool consisting of 86K examples from the GQA training split (Hudson & Manning, 2019), 30K from TextVQA (Singh et al., 2019), 28K from ChartQA (Masry et al., 2022), 10K from AI2D (Kembhavi et al., 2016), and 6K image-context examples from ScienceQA (Lu et al., 2022). The video pool contains 90K examples: 30K from ActivityNet-QA (Yu et al., 2019), 30K from the NExT-QA training split (Xiao et al., 2021), and 30K from TGIF-QA (Jang et al., 2017). We remove corrupted media and duplicate normalized question–visual-input pairs before sampling.

We then reserve 5% of each source as controller validation data and never use these examples for policy updates. V*Bench, HR-Bench, DocVQA, RealWorldQA, MMMU, Video-MME, LongVideoBench, and EgoSchema do not contribute examples to controller training. NExT-QA is the only headline benchmark whose official training split contributes to the controller mixture; its validation and test examples remain held out.

Offline demonstrations. From the remaining training data we construct 20K image and 10K video demonstration trajectories for the warm start described in the method. The teacher uses a frozen SigLIP-SO400M/14 representation (Zhai et al., 2023) as its controller-independent image–text embedding space. For videos, frame embeddings are computed independently and pooled only for teacher scoring. The teacher weights in the memory score are $\alpha_q = 1.0$, $\alpha_\nu = 0.5$, $\alpha_c = 0.25$, and $\alpha_r = 0.5$. We use $\tau_{\mathrm{mem}}^{\mathrm{teach}} = 0.10$ for optional anchor stopping.

For demonstration construction only, we form a finite proposal pool before applying the acquisition teacher. Image proposals use centers on a 5×5 normalized grid with square side lengths in $\{0.125, 0.25, 0.50, 0.75\}$, together with global views. Video proposals use the temporal templates defined in Appendix F.4, with temporal centers quantized to a 16-bin timeline; each temporal proposal is paired with a global view and a 3×3 grid of local crops at side lengths $\{0.25, 0.50\}$. We first rank all budget-feasible proposals using the frozen teacher embedding based on query relevance, novelty, and coverage, and evaluate only the top eight proposals (plus the global proposal when feasible) with the frozen-backbone reference-answer likelihood. Each shortlisted proposal is

Table 3: Training-source rewards and headline evaluation metrics. Training rewards lie in $[0, 1]$; headline metrics are reported on a $[0, 100]$ scale. Only the official NExT-QA training split contributes to both source families.

Source / benchmark	Role	Reward / metric
GQA; ActivityNet-QA	Train	Normalized answer matching
TextVQA	Train	Multi-reference answer agreement
ChartQA	Train	Numeric-tolerant or normalized answer matching
AI2D; ScienceQA	Train	Choice correctness
TGIF-QA	Train	Subtask-specific choice, answer, or count reward
NExT-QA	Train/test	Choice correctness on disjoint official splits
V*Bench	Test	Answer correctness
HR-Bench-4K/8K	Test	Official cyclic-permutation score
DocVQA	Test	ANLS
RealWorldQA	Test	Deterministic option/string matching
MMMU	Test	Official parsing and scoring
Video-MME; LongVideoBench; EgoSchema	Test	Choice accuracy

scored only after teacher-side memory compression under the prescribed B_{mem}. The teacher continues when the best one-step utility improvement exceeds $\tau_{\mathrm{view}}^{\mathrm{teach}} = 0.005$. Proposal discretization and shortlisting are used only to construct demonstrations; the learned WATCH policy retains the continuous budget-feasible view parameterization during on-policy training and inference.

F.4 Observation Operator and Resource Budgets

Cost instantiation. We instantiate the normalized observation cost as

$$c_{\mathrm{obs}}(v) = \frac{\alpha_{\mathrm{enc}} + \alpha_{\mathrm{pix}}|I|\,\mathrm{Area}(R)\rho^2}{\alpha_{\mathrm{enc}} + \alpha_{\mathrm{pix}}}, \qquad \alpha_{\mathrm{enc}} = 0.1, \quad \alpha_{\mathrm{pix}} = 0.9. \tag{74}$$

A single full-frame observation at unit fidelity therefore has cost one. The fixed encoding term accounts for per-view launch and encoder overhead, while the remaining term scales with the amount of visual evidence processed. All observation costs reported for WATCH include the initial preview.

For image views, unit fidelity corresponds to a canonical full-frame raster of

$$P_{\mathrm{ref}}^{\mathrm{img}} = 1280 \times 28^2 = 1{,}003{,}520$$

pixels. For video we use

$$P_{\mathrm{ref}}^{\mathrm{vid}} = 256 \times 28^2 = 200{,}704$$

pixels per full-fidelity frame. These are controlled reference rasters for resource accounting rather than the maximum resolutions supported by the backbone. Before encoder execution, a selected region is cropped from the original media and assigned a requested raster proportional to $\mathrm{Area}(R)\rho^2 P_{\mathrm{ref}}$.

We quantize each requested raster conservatively to the backbone-native visual grid: multiples of 28 for Qwen2.5-VL and multiples of 32 for Qwen3-VL. The quantized dimensions, the available native resolution of the source crop, the backbone's maximum admissible raster, and the remaining observation budget jointly determine the realized upper bound on ρ. The realized cost is recomputed after this quantization and is included in the feasibility mask; therefore raster discretization cannot cause an action to exceed the remaining hard budget. We never upsample a crop beyond its available native pixels.

Spatial and fidelity actions. For global views, $R = R_{\mathrm{full}}$ and $\rho \in [0.25, 1.0]$. Local views use continuously parameterized axis-aligned boxes with width and height in $[0.125, 0.75]$ of the original visual extent. To permit genuine high-fidelity reinspection, local image views use a nominal fidelity

range $\rho \in [0.5, 4.0]$, while local video views use $\rho \in [0.5, 2.0]$. Hence a small region may receive a higher pixel density than the unit-fidelity global view. The effective upper bound is further reduced by the remaining budget, source resolution, and backbone-supported raster as described above. The Beta-distribution parameters of the continuous policy factors are parameterized as $0.5 + \mathrm{softplus}(\cdot)$ and initialized with latent mean 0.5. Prefix masking and budget-dependent support transformations are applied before sampling, so every executed view is feasible without post-hoc projection.

Temporal actions. For images, $I = \{1\}$. For videos, we use even-sized temporal supports compatible with the backbone's temporal patching and let the temporal policy choose among

$$(|I|, s) \in \left\{(2, \tfrac{1}{32}), (4, \tfrac{1}{16}), (4, \tfrac{1}{4}), (8, \tfrac{1}{4}), (8, 1), (16, 1)\right\},$$

where s is the normalized temporal span. For $s < 1$, the window center is selected on a 32-bin normalized timeline and the requested frames are uniformly distributed within the selected span; $s = 1$ denotes sparse whole-video inspection. A local spatial crop is applied consistently to all frames in the selected temporal support. This library provides short local inspection and sparse long-range coverage while keeping the action space small.

Fixed preview. For images, O_0 is one full-frame observation at $\rho_{\mathrm{pre}} = 0.25$. Its visual grid is deterministically reduced by adaptive spatial pooling to at most 32 backbone-compatible tokens, yielding

$$c_{\mathrm{pre}}^{\mathrm{img}} = 0.1 + 0.9(0.25)^2 = 0.15625.$$

For videos, O_0 is a single multi-frame preview view containing eight uniformly spaced frames, each using R_{full} at $\rho_{\mathrm{pre}} = 0.25$. Each frame is deterministically reduced to at most eight backbone-compatible visual tokens, so M_0 contains at most 64 tokens and

$$c_{\mathrm{pre}}^{\mathrm{vid}} = 0.1 + 0.9 \times 8(0.25)^2 = 0.55.$$

Thus the fixed per-view encoding term is charged once for the complete preview view, consistent with the observation-cost definition. The preview is identical across policies sharing the same input and prescribed budget and contains no learned WATCH decision.

Supported budgets. For images we train a single controller over

$$B_{\mathrm{obs}} \in [0.5, 4.0], \qquad B_{\mathrm{mem}} \in \{32, 64, 128, 256\}, \qquad K_{\max} = 4,$$

and use $(2, 128)$ as the default operating point. For videos,

$$B_{\mathrm{obs}} \in [4, 32], \qquad B_{\mathrm{mem}} \in \{64, 128, 256, 512\}, \qquad K_{\max} = 6,$$

with default $(16, 256)$. During full-range training, B_{obs} is sampled log-uniformly and B_{mem} uniformly from the corresponding discrete set. Budget interpolation is evaluated at image observation budgets $\{0.75, 1.5, 3.0\}$ and video budgets $\{6, 12, 24\}$, none of which is used as a curriculum endpoint.

Table 4: Nominal observation costs under Eq. 74. $F = |\mathcal{I}|$ and $A = \mathrm{Area}(R)$. Values precede native-grid quantization; execution uses the realized raster and charges the fixed term once per view.

View	F	A	ρ	c_{obs}
Image preview	1	1	0.25	0.15625
Image global	1	1	1	1.00000
Image local	1	0.0625	2	0.32500
Image local	1	0.0625	4	1.00000
Video preview	8	1	0.25	0.55000
Video local	4	0.25	1	1.00000
Video global	8	1	0.5	1.90000
Video global	16	1	1	14.50000

Full-input reference. Resource ratios are defined relative to a fixed canonical full-input configuration rather than the number of raw frames stored in a video file. For an image, the canonical Full Input processes the complete frame once at $\rho = 1$ and $P_{\text{ref}}^{\text{img}}$; for video it processes 64 uniformly spaced global frames at $\rho = 1$ and $P_{\text{ref}}^{\text{vid}}$ per frame. We denote the resulting encoded-token count by $N_{\text{enc}}^{\text{full}}$ and the corresponding uncompressed visual context by N_{full}, and report

$$r_{\text{obs}} = \frac{N_{\text{enc}}}{N_{\text{enc}}^{\text{full}}}, \qquad r_{\text{ctx}} = \frac{N_{\text{backbone}}}{N_{\text{full}}}.$$

For WATCH and methods whose acquisition path uses the common Qwen visual front-end, the cross-method acquisition ratio used in the main paper satisfies $r_{\text{acq}} = r_{\text{obs}}$. For pipelines with an auxiliary visual selector or encoder, r_{acq} instead measures the canonical-equivalent amount of visual evidence processed by the pre-answer acquisition/selection path, normalized by the canonical Full Input. Thus selector-side full scans are counted even when only a small subset of the resulting evidence is ultimately delivered to the answering backbone.

These canonical references make ratios comparable across inputs and operating points; they do not imply that every physical frame of a source video is decoded. To ensure that the reference raster does not artificially limit the passive baseline on high-resolution tasks, we additionally report an unconstrained high-resolution Qwen baseline in Appendix G.

F.5 Controller and Memory Configuration

Controller architecture. The recurrent evidence state uses a single-layer GRU with hidden dimension $d_h = 512$. Backbone-compatible visual features are projected to $d_c = 256$ for controller-side computation. Spatial–temporal metadata and the resource descriptor are embedded into 64 dimensions each, while the old/new provenance indicator uses a 16-dimensional embedding. The candidate contextualizer G_θ contains two Transformer layers with hidden width 256, eight attention heads, a 1024-dimensional feed-forward block, GELU activations, and dropout 0.1. The sequential anchor selector uses a 256-dimensional selection state. The complete WATCH policy contains approximately 9.7M trainable parameters; the separate training-only potential estimator contains 5.0M parameters and is discarded at deployment.

Memory compression. Novelty is computed against the persistent memory exactly as defined in the method. For geometry-aware redundancy resolution, we use $\tau_{\text{sem}} = 0.90, \quad \tau_{\text{geo}} = 0.20$, with $(\omega_{xy}, \omega_t, \omega_\rho) = (1.0, 0.5, 0.25)$ for video and $\omega_t = 0$ for images. Semantic merge weights use temperature $\tau_m = 0.07$ and $\epsilon = 10^{-6}$ in the log-fidelity discrepancy. Candidates failing either the semantic or geometric compatibility criterion are pruned rather than merged. All thresholds are selected once on the held-out controller validation set and then fixed for every benchmark.

Halting. We select the deterministic inference threshold by sweeping $\tau_{\text{halt}} \in \{0.35, 0.40, \ldots, 0.70\}$ on controller validation examples and maximizing validation utility under the default resource coefficients. This gives $\tau_{\text{halt}} = 0.55$ for images and 0.50 for videos. The same threshold is used across all test benchmarks and observation budgets of the corresponding modality; it is not retuned per dataset.

F.6 Training Protocol

Behavior-cloning warm start. We optimize the demonstration objective for three epochs with AdamW (Loshchilov & Hutter, 2019), learning rate 2×10^{-4}, weight decay 0.01, gradient-norm clipping at 1.0, and a 5% linear warmup followed by cosine decay. Demonstration decisions from the halt, view, and memory policies are sampled in equal proportions when constructing minibatches. The memory-sequence term is averaged over realized anchor-selection rounds as in the training objective. The resulting controller is copied to the fixed reference policy π_{ref} before on-policy training.

On-policy optimization. Each same-context group contains $G = 8$ complete trajectories. We use clipping parameter $\epsilon = 0.2$, potential coefficient $\eta = 0.5$, memory-policy weight $\lambda_{\text{mem}}^{\text{RL}} = 1.0$, KL

coefficient $\beta_{\mathrm{KL}} = 0.01$, and entropy coefficient $\beta_{\mathrm{ent}} = 0.001$. The WATCH policy is optimized with AdamW at learning rate 5×10^{-5}, while the potential estimator uses learning rate 1×10^{-4}. Both use $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.01, gradient clipping at 1.0, 5% warmup, and cosine learning-rate decay. We perform two minibatch epochs for each collected rollout batch and use BF16 throughout.

The resource-regularization coefficients are $(\lambda_{\mathrm{obs}}, \lambda_{\mathrm{mem}}) = (0.03, 0.02)$ for images and $(0.005, 0.02)$ for videos. These values are selected using validation utility and are kept fixed for the main experiments; their sensitivity is evaluated separately in Appendix I.

Budget curriculum. The first 30% of on-policy updates use a curriculum. For images, the lower observation-budget endpoint decreases linearly from 2.0 to 0.5 and the minimum sampled memory capacity decreases from 128 to 32. For video, the corresponding endpoints decrease from 16 to 4 and from 256 to 64. The remaining 70% of updates sample the complete deployment ranges specified above. Importantly, only the sampling distribution changes during the curriculum; feasibility masks, cost definitions, and action supports are identical to those used at inference.

An image run uses 1,000 policy iterations with 32 context groups per iteration, corresponding to 256K sampled trajectories. A video run uses 750 iterations, corresponding to 192K trajectories. At every iteration we first collect trajectories with a snapshot θ_{old}, update the disjoint potential estimator on the resulting continuation targets, freeze the potential, and then update the WATCH policy against θ_{old}. The controller training schedule uses three independent seeds, 17, 29, and 43.

Reward computation and terminal decoding. For each rollout, the frozen answering backbone decodes greedily with temperature 0 and at most 128 new tokens, using the same terminal answer-format instruction as evaluation. Controller actions remain stochastic during rollout collection. Multiple-choice rewards are binary correctness after deterministic answer extraction. For GQA and ActivityNet-QA, Q is normalized exact matching against an accepted reference. String normalization applies Unicode NFKC normalization, lowercasing, whitespace collapse, and removal of surrounding punctuation while preserving internal numeric signs and decimal points. TextVQA uses leave-one-reference-out agreement: for normalized references $a_1, \ldots, a_m$, $Q = m^{-1} \sum_{j=1}^{m} \min\{\frac{1}{3} \sum_{i \neq j} \mathbf{1}[y = a_i], 1\}$. ChartQA uses exact matching for nonnumeric answers and accepts a parsed numeric answer when $|y - a| \leq 0.05|a|$; a zero reference requires a zero answer. Numeric strings are converted to a common unit before comparison. For TGIF-QA, choice and frame-answer examples use correctness and normalized matching, respectively; count examples use $Q = 1/(1 + |n - \widehat{n}|)$, with invalid counts receiving zero. All rewards are bounded in $[0, 1]$ and use no learned judge. The trajectory utility subtracts $\lambda_{\mathrm{obs}} C_{\mathrm{obs}} + \lambda_{\mathrm{mem}} c_{\mathrm{mem}}(M_K)$; costs incurred before a shaping state are excluded from its continuation target. Group normalization is applied subsequently to the stage-specific advantages as specified in Appendix D.3.

F.7 Baseline Implementations and Fairness

Passive baselines. **Full Input** uses the canonical full-input configurations defined above and performs no adaptive visual selection. **Preview Only** invokes the frozen backbone directly on the deterministic WATCH preview M_0. **Uniform Budget** uses the same observation operator, B_{obs}, B_{mem}, and maximum horizon as WATCH but removes query-adaptive view selection: image regions follow a deterministic coverage-preserving spatial schedule, while video timestamps are uniformly allocated over the full duration. When its candidate bank exceeds B_{mem}, visual tokens are reduced by deterministic geometry-preserving pooling. Thus Uniform Budget isolates the benefit of spending a small visual budget *adaptively*, rather than merely spending less.

Visual-token reduction. We evaluate FastV (Chen et al., 2024), SparseVLM (Zhang et al., 2024), and VisionZip (Yang et al., 2025a). When an official implementation does not natively support Qwen2.5-VL, we port the published token-selection or reduction rule to the common Qwen2.5-VL visual-token interface using the released code as the reference implementation, while preserving the original scoring rule and pruning location. We tune retention to the nearest available target in $\{32, 64, 128, 256\}$ for images and $\{64, 128, 256, 512\}$ for videos. These methods are charged for all visual evidence encoded before token reduction. In particular, methods that prune inside the language stack retain the computation of their preceding full-token layers; we therefore report post-

pruning token count together with layerwise token volume, FLOPs, and measured latency rather than treating the final token count as a proxy for total compute.

Additional learned and query-adaptive selectors. TPRL (Cao et al., 2026) uses its compact visual representation and language-guided selection policy, with its demonstration and policy-optimization stages retained. Its inference mask is decoded deterministically. EvoComp (Song et al., 2026) uses its joint visual–text selector and evolutionary subset supervision. SCORE (Wang et al., 2026b) retains its temporal-surprise features, group-based selector training, and deterministic global Top-K inference. EXPLORE-THEN-SELECT (Shi et al., 2025) retains its key/delta-token candidates and query-aware candidate scoring. We use the complete canonical encoded input for these comparisons, preserve original token positions, and keep the answering backbone frozen. Method-specific selector objectives are retained; trainable selectors use the controller-training sources in Appendix F.3, with no headline test examples used for fitting or selection. Retention settings are selected on controller validation data near 128 image tokens or 256 video tokens, and actual retained counts are reported. For threshold-based selectors, the achieved count is measured rather than assumed equal to the target. All candidate-scoring passes and selection overhead are included in system measurements.

Active image perception. For controlled comparisons, we reimplement the visual-search procedures of V* (Wu & Xie, 2024) and ZoomEye (Shen et al., 2025) around the same frozen Qwen2.5-VL-7B-Instruct terminal backbone, following their released code and method-specific internal search prompts. All methods receive the same raw visual input and terminal task question, and we account for every acquired view and every language-backbone invocation in the trajectory. DeepEyes (Zheng et al., 2026) and AdaptVision (Lin et al., 2026) post-train the MLLM itself for active visual reasoning or acquisition and therefore do not admit strictly frozen-backbone counterparts under their original formulations; we evaluate their released Qwen2.5-VL-7B recipes as external references rather than use them for matched-backbone significance claims. Q-Zoom (Shi et al., 2026b) modifies intermediate backbone routing and uses targeted post-training for coarse-to-fine RoI perception; we likewise report it as an additional external reference in Appendix G. For these adaptive references, acquisition statistics are recomputed from the executed visual processing under our canonical accounting rather than copied from their native token or throughput metrics.

Video selection. For VAP (Ma et al., 2026a), observation accounting includes all visual processing performed by the selector, not only the frames ultimately passed to the answering model. VideoTree (Wang et al., 2025c) is evaluated both with its public pipeline and, where possible, by passing its selected visual evidence to the common Qwen2.5-VL terminal backbone. Costs associated with tree construction and intermediate visual processing are retained in the system measurements.

Recent agentic perception systems. Several concurrent systems are closely related conceptually but operate under different inference contracts. ActiveScope (Wang et al., 2026d) focuses on training-free high-resolution seeking and correction, while Active Video Perception (Wang et al., 2026e) and LensWalk (Li et al., 2026) use multi-round MLLM planning and observation for long-video evidence seeking, with explicit evidence-sufficiency reflection in the former. OmniAgent (Xing et al., 2026) maintains persistent textual memory inside an omni-modal agent, whereas HIVE (Tan et al., 2026) first constructs a compact caption-augmented video environment and then performs tool-based spatiotemporal exploration. Because these systems involve heterogeneous reasoners, repeated language-model interaction, textual or bimodal memories, or preconstructed video environments, their reported frame, token, and latency quantities are not directly equivalent to r_{acq} and r_{ctx}. We therefore treat them as complementary system references rather than map their published efficiency numbers into the matched WATCH tables.

Controlled versus published comparisons. Controlled methods use the same raw media, frozen Qwen2.5-VL terminal backbone, terminal task prompt, answer decoding, and evaluation code; method-specific internal search or selection prompts follow the original method. Published-model results are displayed separately for external context and are not used for claims of statistically significant superiority under matched compute. During budget sweeps, an active baseline is assigned the nearest operating point whose measured observation cost does not exceed the target; budget matching uses this measured cost rather than tool-call, selected-frame, or terminal-token counts.

Acquisition ledger. Each visual-processing event records its module, source frame/crop, realized raster, output-token count, and cache provenance. For the common visual front-end, its canonical-equivalent amount is the number of newly encoded backbone-interface tokens divided by $N_{\mathrm{enc}}^{\mathrm{full}}$. For an auxiliary visual encoder, we use the processed raster area divided by the modality-specific canonical full-input raster area, summed over frames. We sum these amounts over executed encoding events to obtain r_{acq}. Reusing cached features introduces no new encoding event; encoding the same pixels again does. Thus a selector scan followed by a separate answering-encoder pass contributes both events. This evidence-volume normalization is accompanied by module-level FLOPs and latency. WATCH's training cost C_{obs} is computed separately from its realized views using Eq. 74.

Table 5: Per-method acquisition and backbone accounting. Shared encoded features are charged once; separately executed encoders are charged separately.

Methods	Acquisition events	Additional computation
Full Input; Preview Only	Complete canonical input; fixed preview, respectively	Terminal backbone execution
Uniform Budget; WATCH	Preview and every executed view, including revisits	Pooling or controller/Retain computation; terminal backbone execution
FastV; SparseVLM; VisionZip	All input encoded before reduction	Selection and all executed language layers at their actual token counts
TPRL; EvoComp; SCORE	Complete canonical visual encoding	Learned selector and retained-token backbone execution
EXPLORE-THEN-SELECT	Canonical encoding, plus any uncached re-encoding	Every candidate-scoring pass and final answering pass
V*; ZoomEye	Every newly encoded search view	All search and answering calls
VAP; VideoTree	Selector-side scans and any separate answering-encoder passes	Selection/tree construction and all intermediate calls
External adaptive models	All executed visual-processing events	Native internal calls and terminal answer generation

F.8 Inference and System Measurement

Task inference. WATCH uses the deterministic policy described in the method: thresholded halting, maximum-probability feasible discrete view choices, means of the budget-feasible continuous distributions, and greedy anchor/STOP decoding. The terminal Qwen backbone uses greedy decoding with temperature 0 and a maximum of 128 generated tokens. Multiple-choice tasks use a fixed answer-format instruction and deterministic answer extraction. All methods receive the same textual prompt and stopping criteria.

Resource accounting. For every example we record cumulative normalized observation cost C_{obs}, total encoded visual tokens N_{enc}, number of adaptive acquisitions K, cumulative candidate volume N_{ctrl}, memory-policy rounds R_{mem}, final backbone visual tokens N_{backbone}, and the number of language-backbone invocations N_{BBcall}. WATCH has $N_{\mathrm{BBcall}} = 1$ by construction; intermediate visual-front-end executions remain included in N_{enc} and latency. For baselines with auxiliary visual encoders or selectors, we additionally record their processed pixels, frames, and wall-clock time.

Latency and memory. Deployment efficiency is measured on a single NVIDIA H100 SXM 80GB GPU with batch size one. Each configuration receives 20 warm-up requests followed by at least 200 measured examples sampled stratified by benchmark. We call `torch.cuda.synchronize()` immediately before and after each timed stage. Time-to-first-token (TTFT) includes media decoding, visual preprocessing, all adaptive acquisition/controller operations, visual encoding, and terminal backbone prefill up to the first generated token. End-to-end latency additionally includes autoregressive generation.

Because natural answers have different lengths, we measure two latency protocols: normal task latency with benchmark-specific stopping and a controlled 64-new-token decode. Peak GPU memory is measured using `torch.cuda.max_memory_allocated()` after resetting peak statistics for

each request. We separately log visual-perception/controller time and terminal backbone time, so controller overhead is not implicitly attributed to the language model.

Compute environment. Training uses two nodes with 16 NVIDIA H100 SXM 80GB GPUs. A standard controller run uses eight GPUs, while the remaining devices parallelize evaluation and independent seeds. A primary image run takes approximately 22 hours and a video run approximately 36 hours on eight GPUs; including the three headline seeds, baseline ports, demonstrations, and core ablations, the full experimental study consumes approximately 4.2K H100 GPU-hours.

F.9 Statistical Protocol

Table 9 reports the mean and standard deviation across three independently trained controllers. Other diagnostic tables report the corresponding seed-averaged quantity unless explicitly stated otherwise.

For comparison with the strongest controlled baseline at each main operating point, we first average the per-example WATCH *scores* across the three controller seeds and then compute a paired cluster bootstrap with 10,000 resamples over underlying question identifiers. For benchmarks containing repeated views, answer-order permutations, or other multiple evaluation instances derived from the same underlying question, all instances sharing that question identifier are resampled together as a single cluster; for ordinary benchmarks, each evaluation example forms its own cluster. We report the percentile 95% confidence interval of the mean paired score difference, WATCH minus baseline. This prevents correlated variants of the same underlying question from being treated as statistically independent observations.

Latency comparisons in the paper report the mean over measured requests. The sample order is fixed across competing methods, and the first 20 requests of every configuration are excluded as warm-up. Hyperparameters, thresholds, and operating points are selected exclusively on the controller validation set and are not changed after inspecting test-set results.

G Full Benchmark Results

This section expands the compact comparison in Table 1. We first report all individual image and video benchmarks under the primary Qwen2.5-VL-7B-Instruct backbone, then evaluate transfer to Qwen3-VL-8B-Instruct, and finally report variation across independent WATCH training seeds. System costs for the same configurations are given in Appendix H.

G.1 Full Qwen2.5-VL-7B Results

Image reasoning. Table 6 separates the two general-reasoning benchmarks that are compressed into the "Gen." column of the main paper. The unconstrained High-Res Full baseline additionally allows the backbone to use its native dynamic-resolution input beyond our canonical unit-fidelity raster. It improves over the controlled Full Input baseline on detail-sensitive tasks, but remains substantially more expensive and still does not recover the gains of query-adaptive high-fidelity acquisition.

The released DeepEyes and AdaptVision checkpoints provide complementary external references for MLLMs explicitly trained to acquire additional visual evidence. Q-Zoom provides a second recent comparison based on lightweight coarse-to-fine gating and query-conditioned high-resolution RoI acquisition. Consistent with its design, Q-Zoom is particularly strong on HR-Bench and DocVQA, whereas WATCH retains the strongest aggregate result by balancing fine-grained acquisition with document and general reasoning performance. WATCH remains slightly below dedicated search policies on V*Bench, but obtains the strongest controlled HR-Bench results. High-Res Full further confirms that the gain is not explained merely by exposing the backbone to more pixels: selective high-fidelity acquisition yields better detail-sensitive performance with substantially less encoded evidence.

Video reasoning. Table 7 gives the corresponding video results. Full-acquisition compression improves terminal context size but does not avoid encoding the canonical 64-frame input. In contrast,

Table 6: Full image results with Qwen2.5-VL-7B-Instruct. High-Res Full is an unconstrained dynamic-resolution reference and lies outside the default controlled observation budget. DeepEyes†, AdaptVision†, and Q-Zoom† use released active/adaptive model recipes rather than the frozen-backbone controlled setting.

Method	V*	HR-4K	HR-8K	DocVQA	RealWorldQA	MMMU	Avg.
Preview Only	63.7	61.4	58.2	83.9	61.4	53.7	63.7
Full Input	71.2	68.8	65.3	93.1	67.9	58.6	70.8
High-Res Full	79.1	71.8	67.9	94.4	68.6	59.0	73.5
Uniform Budget	75.4	70.9	67.8	90.9	67.2	56.9	71.5
FastV	71.5	68.6	65.1	91.2	67.0	57.4	70.1
SparseVLM	71.9	69.2	65.9	92.0	67.5	57.8	70.7
VisionZip	72.6	69.7	66.4	92.6	67.7	58.1	71.2
TPRL	73.0	70.2	67.1	92.8	68.0	58.1	71.5
EvoComp	73.8	71.0	68.0	**93.3**	68.3	58.5	72.2
V*	**91.0**	74.8	71.4	91.7	68.0	57.0	75.7
ZoomEye	90.7	76.1	73.2	92.2	68.4	57.6	76.4
DeepEyes†	90.1	75.1	72.6	93.5	69.5	60.0	76.8
AdaptVision†	89.4	76.8	73.9	92.6	67.3	59.6	76.6
Q-Zoom†	85.9	78.1	76.7	93.9	68.4	59.0	77.0
WATCH	90.2	**78.2**	**75.8**	93.1	**70.0**	**59.0**	**77.7**

WATCH acquires evidence selectively and maintains a bounded terminal memory throughout the trajectory.

Table 7: Full video results with Qwen2.5-VL-7B-Instruct.

Method	Video-MME	LongVideoBench	EgoSchema	NExT-QA	Avg.
Preview Only	52.7	43.8	49.4	71.5	54.4
Full Input	65.8	55.4	60.8	78.9	65.2
Uniform Budget	62.2	50.6	56.9	76.2	61.5
FastV	64.1	52.9	59.5	77.3	63.5
SparseVLM	64.7	53.9	60.3	78.0	64.2
VisionZip	65.0	54.7	60.0	78.4	64.5
SCORE	66.4	56.7	**63.4**	79.8	66.6
EXPLORE-THEN-SELECT	65.7	55.6	62.1	79.2	65.7
VideoTree	63.2	54.4	61.1	75.6	63.6
VAP	65.6	56.0	63.3	80.1	66.3
WATCH	**67.5**	**58.0**	62.9	**81.2**	**67.4**

WATCH gives the strongest aggregate result and improves most clearly on Video-MME and LongVideoBench, where useful evidence may occur far apart in time. VAP and SCORE remain stronger on EgoSchema, consistent with the fact that a specialized long-form frame selector can be particularly effective on individual video distributions. The system comparison in Appendix H shows, however, that the number of frames eventually delivered to the answering model substantially understates the visual processing performed by such selector-based pipelines.

G.2 Transfer to Qwen3-VL-8B

We next replace the primary backbone with Qwen3-VL-8B-Instruct while retaining the same normalized budget definitions and retraining only the lightweight WATCH controller and backbone-specific input projections. Table 8 shows the same qualitative pattern with a stronger backbone. The absolute gain of active perception is smaller because the base model already resolves more visual evidence from a static input, but the efficiency advantage persists.

Table 8: Transfer to Qwen3-VL-8B-Instruct. The stronger backbone narrows the raw-quality gap, while WATCH retains its advantage on evidence-intensive tasks. Selected representative benchmarks are shown for compactness. "Img." and "Vid." are macro-averages over the same six image evaluation settings and four video benchmarks used in Table 1. WATCH results are reported as mean $\pm$ standard deviation over three independently trained controllers.

Backbone	Method	V*	HR-8K	DocVQA	Img.	Video-MME	LongVideoBench	EgoSchema	Vid.
Qwen3-VL-8B	Full Input	82.5	76.5	**95.2**	78.8	72.6	61.8	67.4	71.3
	Uniform Budget	85.0	77.2	93.4	78.9	70.5	59.0	64.2	68.8
	VisionZip	83.3	76.7	94.7	78.8	71.8	60.7	66.6	70.5
	WATCH	**91.6** ± 0.3	**80.4** ± 0.3	95.0 ± 0.2	**81.7** ± 0.2	**73.6** ± 0.2	**63.4** ± 0.3	68.0 ± 0.3	**72.3** ± 0.2

For completeness, a Qwen3-VL VAP run reaches 73.0 on Video-MME, 62.5 on LongVideoBench, 68.6 on EgoSchema, and 83.9 on NExT-QA, yielding a 72.0 four-benchmark average. Thus, the strongest method remains task-dependent even after changing the backbone: VAP retains an advantage on EgoSchema, whereas WATCH provides the strongest aggregate result with substantially lower persistent visual context.

G.3 Seed Variability

Table 9 summarizes variation in task quality and resource usage across the three controller runs.

Table 9: Variation across independently trained WATCH controllers. Image and video scores denote macro-averages over the benchmark suites.

Seed	Image Avg.	Image r_{acq}	Video Avg.	Video r_{acq}
17	77.47	0.305	67.12	0.264
29	77.97	0.316	67.72	0.276
43	77.65	0.309	67.42	0.270
Mean $\pm$ std.	77.69 ± 0.25	0.310 ± 0.006	67.42 ± 0.30	0.270 ± 0.006

G.4 Paired Confidence Intervals

Table 10 reports benchmark-level paired differences and confidence intervals for the comparator set listed in the table, following Appendix F.9.

Table 10: Paired score difference between WATCH and the strongest controlled baseline for each benchmark. Positive values favor WATCH.

Benchmark	Comparator	Δ	95% CI
V*Bench	V*	-0.8	$[-4.2, +1.4]$
HR-Bench-4K	ZoomEye	$+2.1$	$[-1.0, +3.8]$
HR-Bench-8K	ZoomEye	$+2.6$	$[-0.7, +4.3]$
DocVQA	Full Input	-0.1	$[-0.5, +0.3]$
RealWorldQA	ZoomEye	$+1.6$	$[-1.0, +3.6]$
MMMU	Full Input	$+0.2$	$[-0.4, +0.8]$
Video-MME	VAP	$+1.9$	$[+0.4, +2.2]$
LongVideoBench	VAP	$+1.0$	$[+0.6, +1.8]$
EgoSchema	VAP	-0.7	$[-1.5, +0.1]$
NExT-QA	VAP	$+1.1$	$[+0.0, +1.2]$

H Budget and System Efficiency

The terminal retention ratio alone does not characterize WATCH's efficiency: two systems may deliver the same number of tokens to the answering backbone while having processed very different amounts of visual evidence beforehand. We therefore separately report acquisition, terminal context, language-backbone calls, profiled computation, and wall-clock deployment measurements.

H.1 Observation-Budget Sweeps

Table 11 gives the numerical values underlying Figure 2. Increasing the nominal budget does not force WATCH to consume it: once additional observations have low estimated value, the learned policy halts and the realized acquisition cost grows much more slowly than the hard budget.

Table 11: Observation-budget sweeps underlying Figure 2. K is the mean number of adaptive acquisitions after the fixed preview, and $|M_K|$ is the mean terminal memory size. Uniform uses the same realized observation-token envelope as WATCH.

Image, $B_{\text{mem}} = 128$							
B_{obs}	C_{obs}	r_{acq}	K	$\lvert M_K\rvert$	WATCH Avg.	Uniform Avg.	Δ
0.50	0.29	0.10	0.62	55	69.2	66.0	+3.2
0.75	0.34	0.14	0.84	62	72.1	67.9	+4.2
1.00	0.40	0.19	1.09	70	74.8	69.4	+5.4
1.50	0.48	0.25	1.47	77	76.3	70.6	+5.7
2.00	0.56	0.31	1.85	81	77.7	71.5	+6.2
3.00	0.67	0.40	2.04	84	77.8	72.7	+5.1
4.00	0.82	0.52	2.18	86	78.0	73.8	+4.2
Video, $B_{\text{mem}} = 256$							
B_{obs}	C_{obs}	r_{acq}	K	$\lvert M_K\rvert$	WATCH Avg.	Uniform Avg.	Δ
4	1.91	0.08	1.18	132	59.0	56.4	+2.6
6	2.58	0.11	1.52	154	61.7	57.8	+3.9
8	3.38	0.15	1.89	177	64.0	59.4	+4.6
12	4.92	0.21	2.43	199	65.7	60.5	+5.2
16	7.12	0.27	2.91	213	67.4	61.5	+5.9
24	9.68	0.36	3.18	221	67.4	62.4	+5.0
32	12.65	0.48	3.36	226	67.8	63.3	+4.5

H.2 Resource Accounting

Table 12 separates input-side visual processing from backbone-side context. N_{enc} denotes backbone-interface visual tokens and is reported when acquisition uses the common Qwen visual front-end. The cross-method quantity r_{acq} is more general: it additionally accounts for canonical-equivalent visual evidence processed by auxiliary acquisition or selector modules. Accordingly, N_{enc} is left undefined for heterogeneous full-scan selectors such as VAP and VideoTree, while their $r_{\text{acq}} = 1$ reflects processing of the complete canonical candidate stream.

For multi-call image-search methods, a single terminal context is not directly comparable to WATCH; we therefore report the cumulative visual-token volume processed across all language-backbone calls, normalized by N_{full}, as $r_{\text{BB}}^{\text{cum}}$. Relative FLOPs include visual encoding, method-specific selection or controller computation, and language-model prefill, but exclude autoregressive answer decoding. For single-call methods, $r_{\text{BB}}^{\text{cum}} = r_{\text{ctx}}$.

Compression-only methods illustrate why terminal token count alone is insufficient. VisionZip, for example, can match the hard terminal capacity of Uniform Budget, yet still encodes the complete canonical visual input. Conversely, WATCH reduces both sides of the computation. The contrast is also visible for multi-stage image search: V* and ZoomEye may inspect fewer raw visual tokens than Full Input, but repeated language-backbone calls make their cumulative backbone-side processing substantially larger. WATCH invokes the answering backbone exactly once, as required by the inference procedure in Algorithm 1.

The recent adaptive-acquisition references expose a complementary distinction. AdaptVision reduces input-side acquisition but incurs additional MLLM interaction on tool-triggered examples, while Q-Zoom efficiently gates high-resolution RoI processing yet retains a substantially larger answering context than WATCH. WATCH is the only compared adaptive configuration here that combines reduced visual acquisition with a tightly bounded terminal visual context while retaining a single language-backbone call.

Table 12: Detailed resource accounting at the default operating points. Lower is better for all resource columns. For selector-based video methods, r_{acq} includes selector-side visual processing rather than charging only the finally selected frames. N_{enc} is omitted when the selector operates in a visual representation that is not directly comparable to the common Qwen token interface.

Image

Method	N_{enc}	r_{acq}	N_{backbone}	r_{ctx}	N_{BBcall}	$r_{\mathrm{BB}}^{\mathrm{cum}}$	Rel. FLOPs
Full Input	1280	1.00	1280	1.000	1.0	1.000	1.00
High-Res Full	4096	3.20	4096	3.200	1.0	3.200	5.74
Uniform Budget	397	0.31	128	0.100	1.0	0.100	0.23
FastV	1280	1.00	320	0.250	1.0	0.250	0.63
SparseVLM	1280	1.00	286	0.223	1.0	0.223	0.52
VisionZip	1280	1.00	128	0.100	1.0	0.100	0.44
V*	947	0.74	–	–	7.8	3.180	2.91
ZoomEye	845	0.66	–	–	6.2	2.360	2.28
DeepEyes†	742	0.58	–	–	4.1	1.520	1.76
AdaptVision†	448	0.35	–	–	1.4	0.460	0.79
Q-Zoom†	550	0.43	474	0.370	1.0	0.370	0.72
WATCH	397	0.31	81	0.063	1.0	0.063	0.27

Video

Method	N_{enc}	r_{acq}	N_{backbone}	r_{ctx}	N_{BBcall}	$r_{\mathrm{BB}}^{\mathrm{cum}}$	Rel. FLOPs
Full Input	16384	1.00	16384	1.000	1.0	1.000	1.00
Uniform Budget	4424	0.27	256	0.016	1.0	0.016	0.21
FastV	16384	1.00	3072	0.188	1.0	0.188	0.69
SparseVLM	16384	1.00	2048	0.125	1.0	0.125	0.56
VisionZip	16384	1.00	256	0.016	1.0	0.016	0.47
VideoTree	–	1.00	3072	0.188	1.0	0.188	0.45
VAP	–	1.00	2048	0.125	1.0	0.125	0.58
WATCH	4424	0.27	213	0.013	1.0	0.013	0.23

H.3 Deployment Latency and Memory

Table 13 reports the single-H100, batch-size-one measurements specified in Appendix F. We report perception/control time separately from overall TTFT and E2E latency so that the controller overhead of WATCH is not hidden. The controlled 64-new-token protocol is used for the E2E column.

Uniform Budget is slightly faster than WATCH at the same acquisition scale because it incurs no closed-loop controller overhead. At the default image point, WATCH adds only 90 ms of E2E latency while improving average quality by 6.2 points; the corresponding video overhead is 200 ms for a 5.9-point gain.

H.4 Iso-Quality and Iso-Latency Comparisons

Raw speedup factors can obscure differences in achieved task quality. We therefore additionally compare methods at matched quality and latency, following the iso-efficiency view commonly used for adaptive video selection. Table 14 reports the minimum latency required to reach a fixed quality target, while Table 15 reports the best quality achievable under a fixed latency envelope.

At an image-average target of 76.0, WATCH reaches the target at an intermediate observation budget with 1.42 s E2E latency, compared with 5.87 s for DeepEyes and 8.12 s for ZoomEye. On video, WATCH reaches a 66.0 average in 4.11 s, whereas VAP requires 7.15 s. These comparisons avoid interpreting context compression itself as a wall-clock speedup.

H.5 Interaction Between Observation and Memory Budgets

Figure 5 jointly varies the two resource constraints. The resulting pattern confirms that observation and memory are complementary bottlenecks. When observation is extremely limited, increasing

Table 13: Single-H100 deployment efficiency. Perception includes media preprocessing, visual encoding, selection, and controller computation; for multi-call active-search methods it also includes intermediate search-model calls.

Image					
Method	Perception / control (s)	TTFT (s)	E2E (s)	Peak memory (GB)	Avg. quality
Full Input	0.48	2.64	3.22	29.4	70.8
High-Res Full	1.02	5.58	6.17	39.7	73.5
Uniform Budget	**0.28**	**0.96**	**1.54**	19.1	71.5
FastV	0.49	1.98	2.55	23.6	70.1
SparseVLM	0.50	1.91	2.48	22.8	70.7
VisionZip	0.56	1.73	2.31	20.0	71.2
V*	7.81	8.68	9.34	25.8	75.7
ZoomEye	6.71	7.49	8.12	24.9	76.4
DeepEyes†	4.48	5.18	5.87	25.5	76.8
WATCH	0.42	1.09	1.63	**18.8**	**77.7**

Video					
Method	Perception / control (s)	TTFT (s)	E2E (s)	Peak memory (GB)	Avg. quality
Full Input	3.12	9.72	10.51	44.8	65.2
Uniform Budget	**1.42**	**3.12**	**3.91**	22.1	61.5
FastV	3.10	8.24	9.03	36.2	63.5
SparseVLM	3.15	7.52	8.32	32.8	64.2
VisionZip	3.35	6.88	7.66	27.5	64.5
VideoTree	3.55	5.43	6.21	26.3	63.6
VAP	4.42	6.37	7.15	28.7	66.3
WATCH	1.78	3.34	4.11	**21.8**	**67.4**

Table 14: Iso-quality comparison. We report the minimum measured E2E latency required to reach the target macro-average. A dash indicates that the method does not reach the target over its evaluated operating range.

Method	Image Avg. ≥ 76.0 min E2E (s)	Video Avg. ≥ 66.0 min E2E (s)
Uniform Budget	–	–
SparseVLM	–	–
VisionZip	–	–
V*	–	–
ZoomEye	8.12	–
DeepEyes†	5.87	–
VideoTree	–	–
VAP	–	7.15
WATCH	**1.42**	**4.11**

memory has little effect because the relevant evidence was never acquired. Conversely, under a very small memory capacity, increasingly aggressive acquisition quickly saturates because newly observed evidence cannot remain available to the final reasoner.

I Ablations and Training Diagnostics

This section expands the compact decomposition in Table 2. We first isolate the internal representations and control mechanisms, then examine hard feasibility and budget interpolation, analyze how demonstration warm-start and on-policy optimization contribute to training, and finally separate the benefit of active perception from that of reinforcement learning itself.

I.1 Component Ablations

Table 16 removes one design choice at a time while leaving all other training and deployment settings unchanged. The first block reproduces the high-level Acquire–Retain–Halt decomposition from the

Table 15: Iso-latency comparison. Each entry is the best macro-average achieved without exceeding the specified E2E latency envelope.

Method	Image $\leq$ 2.5s	Video $\leq$ 8.0s
Uniform Budget	73.8	63.3
SparseVLM	70.7	–
VisionZip	71.2	64.5
VideoTree	–	63.6
VAP	–	66.3
WATCH	**78.0**	**67.8**

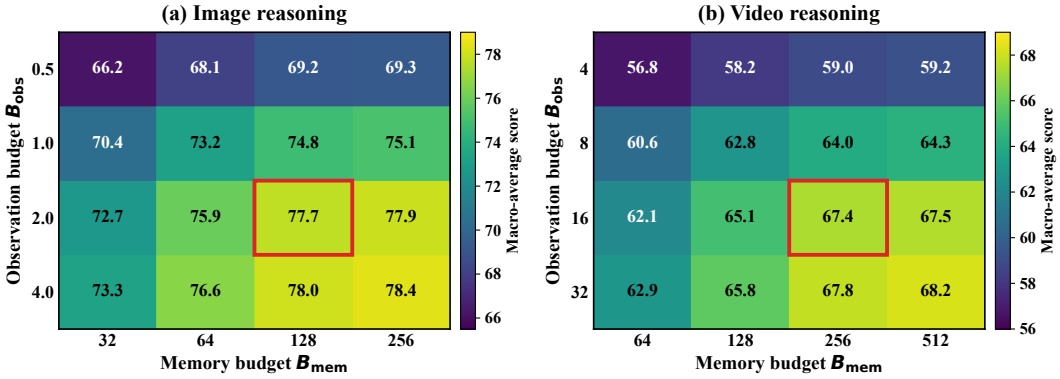


Figure 5: **Interaction between observation and memory budgets.** Macro-average task quality is shown for independently varied $(B_{\text{obs}}, B_{\text{mem}})$. Boxes indicate the default operating points used in the main comparison. Performance saturates along either axis when the other resource becomes the dominant bottleneck, demonstrating that acquisition capacity and persistent visual memory are not interchangeable.

main paper; the remaining rows isolate the memory representation, feasible action parameterization, and learning procedure.

Table 16: Full component ablation at the default operating points. Δ is the displayed-score difference from WATCH; positive values indicate an improvement and negative values a degradation.

Variant	Image Avg.	Δ	Video Avg.	Δ	Primary effect
WATCH	77.7	–	67.4	–	–
Uniform Acquire	72.8	-4.9	62.4	-5.0	query-adaptive evidence acquisition
Static Retain	75.2	-2.5	64.1	-3.3	task-conditioned bounded memory
No Halt	77.8	$+0.1$	67.6	$+0.2$	economical early stopping
w/o memory-relative novelty	76.6	-1.1	66.2	-1.2	repeated evidence
w/o spatial–temporal metadata	76.2	-1.5	65.4	-2.0	location / temporal ambiguity
w/o geometry constraint in merge	76.7	-1.0	65.9	-1.5	cross-region / cross-time merging
Prune only (no merge)	76.9	-0.8	66.5	-0.9	loss of redundant evidence aggregation
Independent Top-K retention	76.3	-1.4	65.8	-1.6	loss of sequential complementary selection
Unconstrained view + repair	76.7	-1.0	66.2	-1.2	train–deploy feasibility mismatch
Terminal reward only	76.5	-1.2	66.1	-1.3	delayed credit assignment
No demonstration warm start	75.8	-1.9	65.3	-2.1	inefficient early exploration
No budget curriculum	76.8	-0.9	66.0	-1.4	weaker optimization at budget extremes

The representation ablations exhibit the expected modality asymmetry. Removing metadata or the geometry constraint is more damaging on video, where visually similar evidence can originate from distinct times and locations. In contrast, disabling feature-space merging causes only a modest loss, consistent with merging being an optional redundancy-resolution mechanism rather than a requirement of the overall WATCH formulation. Sequential anchor selection is more important: independent Top-K retention cannot condition later choices on already selected anchors and therefore preserves less complementary evidence.

I.2 Feasible Action Support

A central design choice in WATCH is to parameterize the view policy so that its support itself satisfies the remaining observation budget, rather than sampling an unconstrained action and repairing it afterwards. Table 17 reports both the resulting pre-repair violation rate and downstream task quality.

Table 17: Hard feasibility of the view policy. Violation is measured before any repair or clipping.

	Pre-repair violation		Task Avg.	
Policy	Image	Video	Image	Video
Unconstrained + repair	13.6%	19.4%	76.7	66.2
Feasible-support WATCH	**0.0%**	**0.0%**	**77.7**	**67.4**

The zero violation rate of WATCH is structural rather than empirical: every sampled discrete prefix admits a feasible continuous completion and the remaining continuous support is budget constrained by construction. The comparison additionally shows that post-hoc repair is not innocuous; it changes the action actually executed by the environment and slightly degrades the learned policy.

I.3 Budget Interpolation

WATCH is trained as a single budget-conditioned controller rather than as a collection of operating-point-specific models. Table 18 evaluates observation budgets that are not used as curriculum endpoints and compares the shared controller with specialists trained only at the corresponding $(B_{\mathrm{obs}}, B_{\mathrm{mem}})$ pair.

Table 18: Interpolation across observation budgets. The memory capacity is fixed at 128 for images and 256 for videos. The shared WATCH controller remains close to separately trained budget-specific specialists.

Image					**Video**				
B_{obs}	0.75	1.50	3.00	Mean gap	B_{obs}	6	12	24	Mean gap
Budget-specific specialist	72.3	76.5	77.9	–	Budget-specific specialist	61.9	65.9	67.5	–
Shared WATCH	72.1	76.3	77.8	-0.17	**Shared WATCH**	61.7	65.7	67.4	-0.17

The small interpolation gap indicates that explicit resource conditioning is sufficient to support a continuum of deployment regimes without retraining a separate policy for each observation budget.

I.4 Training Sample Efficiency and the Role of Demonstrations

Figure 6 compares full WATCH with two on-policy training controls. *Terminal reward only* removes the potential-based dense credit while retaining the same trajectory objective and demonstration warm-start, whereas *No warm start* initializes the controller without behavior cloning but otherwise uses the complete WATCH training procedure. For the warm-started variants, the point at zero on-policy trajectories is therefore the Phase-I behavior-cloning checkpoint before any reinforcement learning update.

Potential shaping is therefore most useful as a credit-assignment mechanism rather than as an alternative task objective. This empirical behavior complements the policy-invariance argument in the method: dense credit reaches a given utility with fewer rollouts, while the terminal task objective remains unchanged.

To distinguish initialization from final policy quality, we additionally evaluate behavior cloning as a complete training recipe. *BC only* freezes the controller immediately after the Phase-I demonstration warm-start in Appendix D.2 and applies the same deterministic inference procedure as WATCH, with no subsequent on-policy updates. The demonstration teacher, reference answers, and teacher-side scoring functions are not used at evaluation time. For comparison, *RL from scratch* is the no-warm-start variant from the component ablation, and *BC + RL, terminal reward* retains the warm-start and on-policy optimization but removes potential-based dense credit.

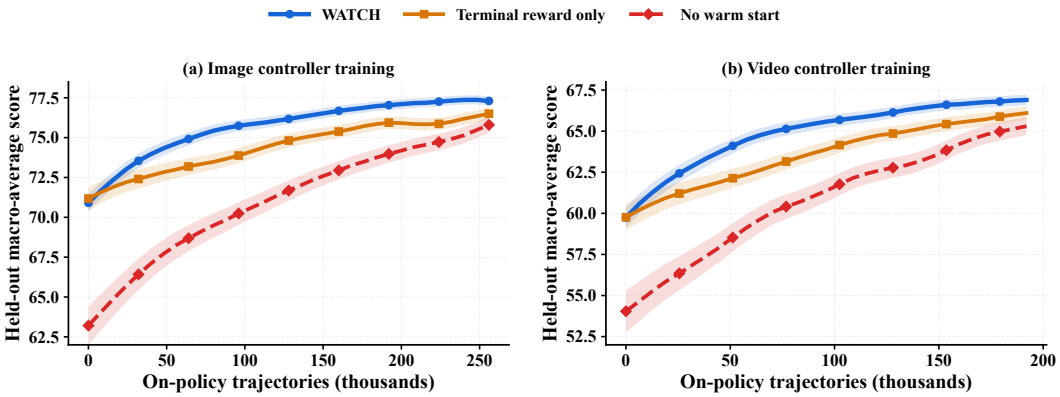


Figure 6: **Training sample efficiency.** Curves report controller-validation quality as a function of sampled on-policy trajectories; solid lines show three-seed means and shaded regions show one standard deviation. For WATCH and *Terminal reward only*, the point at zero on-policy trajectories is the Phase-I behavior-cloning checkpoint before any reinforcement-learning update. Potential-based dense credit accelerates optimization, while the demonstration warm-start avoids the low-quality exploration regime at the beginning of training.

Table 19 shows that the demonstrations provide a useful structured initialization but do not account for the final performance of WATCH. Freezing the BC policy leaves a 4.1-point image gap and a 4.6-point video gap to the full model, while also consuming more visual acquisition and terminal context. Conversely, on-policy training from a random initialization recovers much of the quality but remains 1.9 and 2.1 points below WATCH on image and video reasoning, respectively. Combining the warm-start with on-policy optimization closes most of this gap, and potential-based dense credit provides the remaining improvement while substantially accelerating convergence in Figure 6. Thus the demonstrations primarily provide an exploration prior, whereas the final policy is shaped by downstream on-policy optimization.

Table 19: **Separating behavior cloning and on-policy reinforcement learning.** All variants use the same controller architecture, hard deployment budgets, and deterministic inference protocol at the default operating points. *BC only* is the Phase-I behavior-cloning policy with no on-policy reinforcement learning; *RL from scratch* omits the demonstration warm-start; and *BC + RL, terminal reward* performs on-policy optimization from the BC checkpoint without potential-based dense credit. Resource columns report the realized acquisition and terminal-context usage under the shared hard budgets, without post-hoc matching across variants.

	Image			Video		
Training recipe	Avg.↑	r_{acq} ↓	r_{ctx} ↓	Avg.↑	r_{acq} ↓	r_{ctx} ↓
BC only	73.6	0.35	0.076	62.8	0.31	0.015
RL from scratch	75.8	0.33	0.068	65.3	0.29	0.014
BC + RL, terminal reward	76.5	0.32	0.065	66.1	0.28	0.014
WATCH	**77.7**	**0.31**	**0.063**	**67.4**	**0.27**	**0.013**

I.5 Does Reinforcement Learning Alone Explain the Gain?

The preceding experiment asks whether demonstration imitation alone is sufficient. We next ask whether reinforcement learning itself, rather than iterative active perception, could explain the gain. Inspired by the Direct RFT control used to separate active sensing from direct reinforcement fine-tuning in ACTIVE-O3 (Zhu et al., 2026), we construct a control compatible with our frozen-backbone setting.

Static Controller-RL receives the complete canonical visual input once and trains a one-shot stochastic memory selector with the same controller data, terminal task reward, group size, and policy op-

timizer as WATCH. It has no iterative Acquire or Halt decisions. Thus it can learn task-conditioned retention through RL, but cannot actively request missing high-fidelity evidence. The resulting comparison with Frozen Full Input and WATCH is reported in Table 20.

Table 20: RL without iterative active perception on image reasoning. "HR Avg." is the mean of HR-Bench-4K/8K and "Gen. Avg." is the mean of RealWorldQA/MMMU.

Method	HR Avg.	Gen. Avg.	r_{acq}	r_{ctx}
Frozen Full Input	67.1	63.3	1.00	1.000
Static Controller-RL	69.2	<u>63.8</u>	1.00	0.100
WATCH	**77.0**	**64.5**	**0.31**	**0.063**

Static Controller-RL modestly improves over the frozen static baseline, showing that RL can learn useful task-conditioned retention. WATCH improves high-resolution reasoning by 7.8 points over Static Controller-RL. Together with the Acquire ablation in Table 2, this control shows that the gain is not explained by reinforcement learning alone and is driven primarily by active evidence acquisition.

Preview-conditioned open-loop acquisition. The Open-loop Planner selects a variable-length sequence of feasible views using only the question, fixed preview, prescribed budgets, and preceding planned actions. The entire sequence and its stopping position are committed before acquiring additional evidence. Planned costs update the remaining budget, with the same raster checks and maximum horizon as WATCH. A recurrent Retain module compresses each resulting candidate bank under the same memory cap, and the frozen backbone answers once from the terminal memory. Newly acquired features are available to Retain and the final answer, while the planner's inputs remain fixed to its preview-conditioned planning history. The planner and Retain module share a joint training protocol using the same data mixture, trajectory counts, reward coefficients, and validation protocol as WATCH, with the same three seeds. This control tests the value of evidence-dependent acquisition and stopping beyond a learned query-conditioned acquisition schedule.

Table 21: Preview-conditioned planning versus closed-loop acquisition at the default budgets. Score differences are WATCH minus Open-loop Planner.

Suite	Method	Avg.	r_{acq}	r_{ctx}
Image	Open-loop Planner	75.8	0.49	0.092
	WATCH	77.7	0.31	0.063
	Score difference	+1.9	-0.18	-0.029
Video	Open-loop Planner	65.7	0.58	0.015
	WATCH	67.4	0.27	0.013
	Score difference	+1.7	-0.31	-0.002

The paired evaluation first averages each example's score across the three seeds. Within each benchmark, we resample question clusters with replacement, keeping all answer-order variants together and using identical resampled clusters for both methods. Settings sharing underlying questions use shared resampling weights. We compute each benchmark's paired difference and then average equally across the six image settings or four video benchmarks. The 2.5th and 97.5th percentiles of 10,000 replicates form the interval. This quantifies example-sampling uncertainty conditional on the evaluated trained policies.

I.6 Resource-Regularization and Preview Sensitivity

Table 22 varies the two soft resource coefficients while retaining the same hard deployment budgets. As expected, stronger observation regularization shifts the policy toward lower acquisition, while stronger memory regularization reduces the terminal context. Performance changes smoothly rather than depending on a narrow coefficient setting.

The memory thresholds are similarly stable: $\tau_{\mathrm{sem}} \in \{0.85, 0.90, 0.95\}$ changes the image/video macro-average by at most 0.3/0.4 points, and $\tau_{\mathrm{geo}} \in \{0.10, 0.20, 0.30\}$ by at most 0.4/0.5. Finally,

Table 22: Sensitivity to the resource-regularization coefficients. Multipliers are relative to the modality-specific default coefficients reported in Appendix F.

Parameter	Mult.	Image Avg.	Image r_{acq}	Image r_{ctx}	Video Avg.	Video r_{acq}	Video r_{ctx}
λ_{obs}	$0.5\times$	77.6	0.38	0.066	67.2	0.34	0.014
	$1.0\times$	77.7	0.31	0.063	67.4	0.27	0.013
	$2.0\times$	76.6	0.24	0.060	66.3	0.21	0.012
λ_{mem}	$0.5\times$	77.4	0.32	0.075	67.0	0.28	0.015
	$1.0\times$	77.7	0.31	0.063	67.4	0.27	0.013
	$2.0\times$	76.8	0.30	0.050	66.5	0.27	0.010

preview fidelities $\rho_{\mathrm{pre}} \in \{0.15, 0.25, 0.35\}$ yield image averages $76.8/77.7/77.4$ and video averages $66.2/67.4/67.0$. We retain $\rho_{\mathrm{pre}} = 0.25$: the stronger preview provides no additional quality while consuming more of the observation budget before adaptive perception begins.

J Theory-Aligned Analyses

The theoretical results characterize an optimal budgeted policy and do not imply that a finite neural controller must satisfy the same properties exactly. We therefore use the analyses below as empirical consistency checks rather than as experimental proofs of the propositions.

J.1 Counterfactual Nested Halting

For a visited evidence state $z_k = (h_k, M_k, L_k)$, we hold the evidence fixed and counterfactually vary only the remaining observation budget supplied through the resource descriptor. Let

$$\widehat{h}(z, b) = \mathbb{I}[p_{\mathrm{halt}}(z, \beta(b)) \geq \tau_{\mathrm{halt}}] .$$

For $b_1 < b_2$, we record a nested-halting violation when $\widehat{h}(z, b_2) = 1$, $\widehat{h}(z, b_1) = 0$, i.e., a state that would halt with more remaining budget unexpectedly switches back to continuation after the available budget is reduced.

We evaluate all ordered budget pairs on 10K held-out evidence states per modality. The single-budget variant is trained only at the default operating point but retains the same resource descriptor at inference, making counterfactual budget values out of distribution. In this intervention, only the remaining-budget coordinate $b/B_{\mathrm{obs}}^{\max}$ is changed; the prescribed total budget B_{obs}, memory capacity, evidence state, and remaining horizon are held fixed. The result is shown in Table 23.

Table 23: Nested-halting violations under counterfactual remaining budgets. Lower is better. These results test empirical consistency with the nested stopping structure rather than proving monotonicity of the learned controller.

Policy	Image	Video
Single-budget training	10.9%	13.6%
No budget curriculum	3.9%	5.8%
WATCH	**1.7%**	**2.2%**

The small but non-zero violation rate is expected: the proposition describes an optimal value function, whereas WATCH amortizes the decision with a finite learned controller. Nonetheless, full-range budget-conditioned training produces substantially more consistent stopping behavior than a policy trained at only one operating point.

J.2 Empirical Continuation Advantage

We next ask whether the deployed Halt decision approximately separates states where another perception step is useful from states where it is not. On 2K held-out states per modality, we estimate

immediate halt utility,

$$\widehat{H}(z) = \widehat{Q}(q, M) - \lambda_{\text{mem}} c_{\text{mem}}(M),$$

by answering from the current memory, and estimate one-step continuation utility as

$$\widehat{C}_1(z) = \frac{1}{J} \sum_{j=1}^{J} \left[Q^{(j)} - \lambda_{\text{obs}} c_{\text{obs}}(v^{(j)}) - \lambda_{\text{mem}} c_{\text{mem}}(M_+^{(j)}) \right], \qquad J = 4,$$

where each counterfactual sample suppresses Halt for one step, draws a feasible view $v^{(j)} \sim \pi_\theta^{\text{view}}(\cdot \mid z, b)$, applies the learned memory update, and then answers immediately. We define

$$\widehat{A}_{\text{cont}}(z) = \widehat{C}_1(z) - \widehat{H}(z).$$

This is a tractable one-step proxy rather than the full optimal continuation value in the Bellman recursion. Figure 7 summarizes both the counterfactual halting behavior and the resulting continuation-advantage distributions.

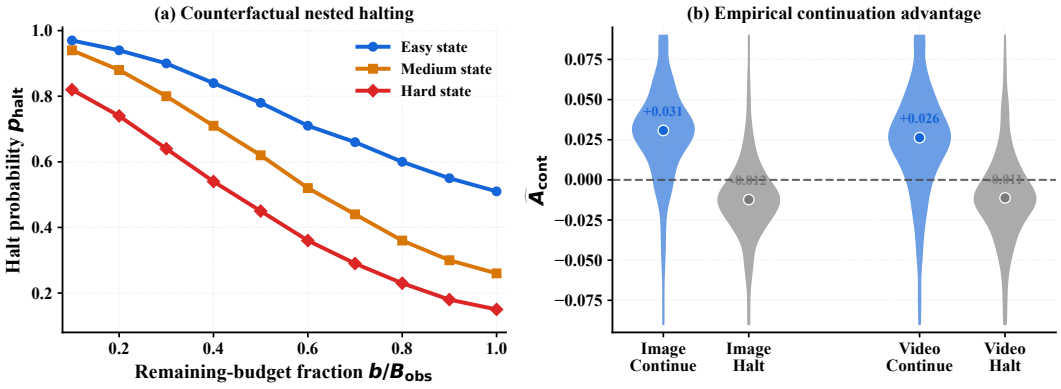


Figure 7: **Behavior of the learned Halt policy. Left:** counterfactual halt probability for representative evidence states while varying only the remaining observation budget; lower budgets consistently increase the tendency to stop. **Right:** empirical one-step continuation advantage for states where WATCH continues versus halts. Continued states concentrate above zero, whereas halted states concentrate below zero. The analysis is an empirical consistency check with budgeted optimal stopping, not an exact solution of the Bellman recursion. For visualization, the representative states are drawn from the easy, medium, and hard preview-NLL terciles defined in Appendix J.4.

For images, the median continuation advantage is $+0.031$ for states where WATCH continues and -0.012 for states where it halts; the corresponding video medians are $+0.026$ and -0.011. The sign of $\widehat{A}_{\text{cont}}$ agrees with the deployed continue/halt decision for 82.9% of image states and 80.7% of video states. The imperfect separation is again expected because the controller predicts long-run continuation value from lightweight state summaries while the diagnostic uses a finite one-step counterfactual.

J.3 Query-Conditioned Rate–Distortion

The memory interpretation in the method suggests that a good bounded memory should preserve the answer distribution induced by the candidate evidence, rather than merely retain visually diverse tokens. We estimate a tractable teacher-forced proxy to this distortion along the reference answer:

$$\widehat{D}_q(C, M) = \frac{1}{T} \sum_{t=1}^{T} D_{\text{KL}}[p_\phi(\cdot \mid q, C, y^*_{<t}) \,\|\, p_\phi(\cdot \mid q, M, y^*_{<t})].$$

This diagnostic measures token-distribution distortion along the reference-answer prefixes. The sequence-level rate–distortion objective and utility bound in Appendix C.5 concern the full answer distribution; the teacher-forced proxy provides an empirical check of answer-relevant memory preservation.

We evaluate 1K candidate states per modality and compare WATCH with random retention, query-relevance Top-K, query-independent semantic merging, and a VisionZip-style token merging rule (Yang et al., 2025a). All methods receive the same candidate bank and the same hard memory capacity. Figure 8 shows the distortion curves across memory capacities, while Table 24 reports distortion and downstream quality at the default capacities.

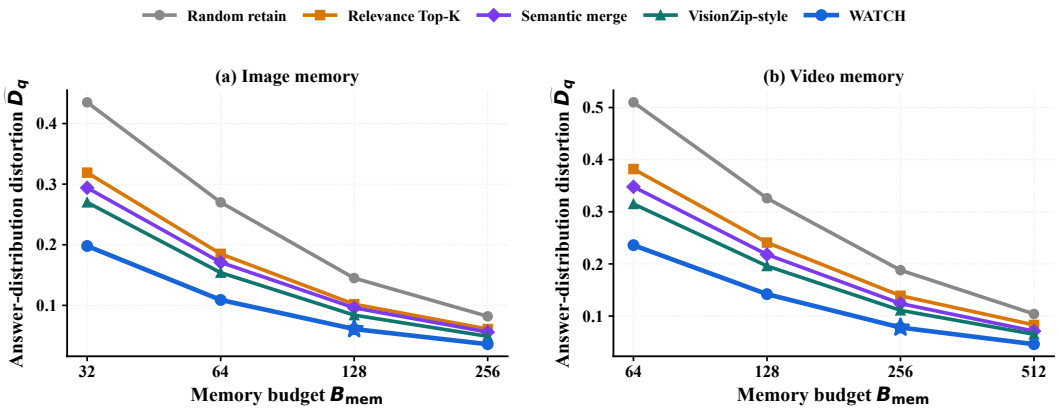


Figure 8: **Teacher-forced query-conditioned answer-distribution distortion.** Lower is better. WATCH preserves the answer distribution induced by the complete candidate evidence more faithfully across memory capacities, with the largest advantage under the tightest memory constraints. Stars mark the default memory capacities used in the main experiments.

Table 24: Distortion and downstream quality at the default memory capacities ($B_{\text{mem}} = 128$ for images and 256 for videos).

	Image		Video	
Retention	$\widehat{D}_q\downarrow$	Avg.↑	$\widehat{D}_q\downarrow$	Avg.↑
Random retain	0.145	73.9	0.188	62.8
Relevance Top-K	0.102	75.4	0.139	64.6
Semantic merge	0.096	75.8	0.124	65.1
VisionZip-style	0.084	76.1	0.111	65.4
WATCH	**0.061**	**77.7**	**0.078**	**67.4**

The ordering in answer-distribution distortion closely tracks downstream task utility. In particular, visually plausible query-independent merging preserves substantially less answer-relevant information at tight capacities, supporting the task-conditioned rather than purely token-count-based interpretation of Retain.

J.4 Input-Adaptive Computation

Finally, we examine whether Halt adapts computation to input difficulty. For analysis only, we compute the length-normalized reference-answer negative log-likelihood under the fixed preview and form difficulty terciles separately within each benchmark. Examples from the corresponding within-benchmark terciles are then pooled across the image or video suite. Lower preview NLL defines an easier example. The difficulty signal is never exposed to WATCH. Table 25 reports the resulting acquisition depth and realized observation-budget usage.

Hard examples trigger roughly 4.5 times as many adaptive acquisitions as easy images and 3.4 times as many as easy videos. Thus the learned Halt head implements input-adaptive computation rather than simply learning a dataset-level fixed stopping depth.

Table 25: Adaptive computation by preview difficulty at the default budgets. K is the number of adaptive acquisitions after the fixed preview.

	Image		Video	
Difficulty	Mean K	$C_{\text{obs}}/B_{\text{obs}}$	Mean K	$C_{\text{obs}}/B_{\text{obs}}$
Easy	0.68	0.17	1.34	0.28
Medium	1.79	0.27	2.83	0.43
Hard	3.08	0.40	4.56	0.63
All	1.85	0.28	2.91	0.45

K Qualitative Results and Failure Cases

Quantitative results describe the aggregate resource trade-off but do not reveal how WATCH constructs evidence over time. We therefore inspect representative trajectories, memory transitions, and failure cases.

K.1 Qualitative Trajectories and Memory Dynamics

Figures 9 and 10 complement the aggregate results with example-level traces of the WATCH controller. The first figure follows complete image and video trajectories through Preview–Acquire–Retain–Halt, while the second magnifies a single Retain update to expose how old and newly acquired evidence compete under the persistent-memory constraint.

Closed-loop behavior. Figure 9 illustrates the distinction between merely reducing terminal context and actively controlling what is observed. In the image example, the global preview provides enough evidence to identify where to inspect but not enough resolution to read the number on the small sign, causing WATCH to refine the same task-relevant region rather than process the entire image at uniformly high fidelity. In the video example, the first targeted window establishes the reference action, while the second concentrates on the immediately following transition needed by the query. The controller then halts with unused acquisition capacity instead of executing all affordable views. The matched Uniform Budget traces illustrate the complementary failure: similar aggregate visual expenditure does not guarantee that the answer-relevant spatial detail or temporal transition is actually observed.

What survives acquisition? Figure 10 shows that Acquire and Retain solve different problems. A newly observed candidate can have high memory-relative novelty and become a new anchor, can be absorbed into an already retained anchor when it contributes redundant compatible evidence, or can be removed when its utility does not justify persistent capacity. Importantly, old and new evidence compete in the same candidate bank: previously retained tokens are not protected merely because they were observed earlier. The resulting memory therefore reflects the current query and the accumulated evidence rather than the chronological order in which visual observations were acquired.

K.2 Failure Modes

We manually inspect 200 incorrect predictions sampled equally from the image and video evaluation suites and assign each example to its dominant failure source. The taxonomy is descriptive rather than exhaustive; ambiguous examples are assigned according to the first failure that prevents the correct evidence from reaching the final reasoner. Table 26 summarizes the resulting taxonomy.

The error distribution also clarifies the scope of WATCH. The controller can adaptively allocate perception, but it cannot recover evidence that is systematically absent from the admissible observation space, nor can bounded visual memory fully replace high-level interleaved reasoning when a task requires a long sequence of dependent visual operations.

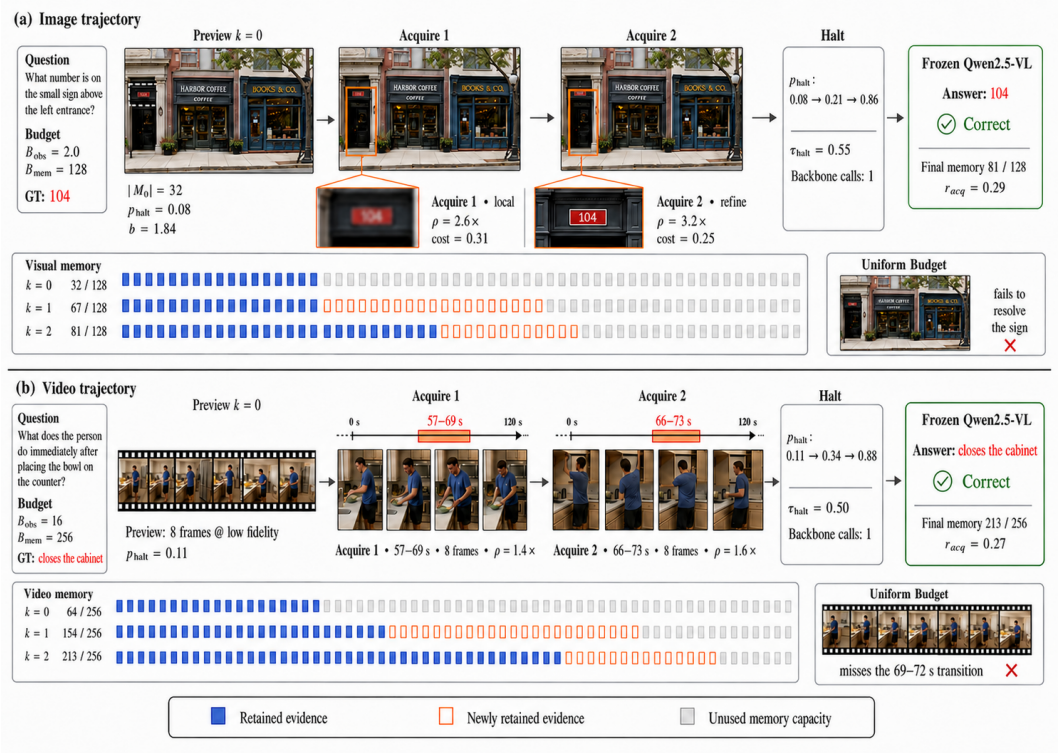


Figure 9: **Representative WATCH trajectories. Top:** on a high-resolution image, the low-cost preview localizes a potentially relevant region but does not resolve the queried fine-grained detail. WATCH first acquires a localized view and then increases fidelity only where additional evidence remains useful. **Bottom:** on a long video, a sparse preview identifies a candidate temporal neighborhood; two targeted windows subsequently resolve the queried event transition. In both examples, the visual memory remains bounded as new evidence is incorporated, the Halt probability rises only after the answer-relevant evidence becomes sufficient, and the frozen answering backbone is invoked once at termination. The memory glyphs summarize the evolution of persistent visual evidence rather than depicting every visual token individually; numerical labels report the actual memory size for the illustrated trajectory. The displayed r_{acq} values are per-example realized acquisition ratios rather than dataset averages.

Table 26: Manual taxonomy of 200 WATCH errors.

Failure source	Share	Typical behavior
Premature halting	26.5%	controller stops before a low-salience detail is inspected
Sparse temporal miss	22.0%	transition requires denser temporal continuity than selected windows
Dispersed evidence under tight memory	18.5%	several distant clues compete for insufficient persistent capacity
Ambiguous preview / query guidance	15.5%	preview does not reveal where useful high-fidelity evidence lies
Incorrect redundancy merge	10.0%	visually similar but logically distinct evidence is merged
Other	7.5%	backbone reasoning or answer-generation errors after adequate evidence

K.3 OOD Stress Test on ActiveVision

ActiveVision (Zhang et al., 2026) is a particularly challenging out-of-distribution test because its 17 tasks explicitly require repeated visual observation across distributed scanning, sequential traversal, and visual attribute transfer. We evaluate the released 85-item benchmark without using any ActiveVision example during controller training.

As shown in Table 27, WATCH substantially improves over static and uniform perception but still remains below the best reported tool-free frontier MLLM and dramatically below human performance. Qualitatively, its gains are concentrated on distributed scanning and visual attribute transfer, whereas

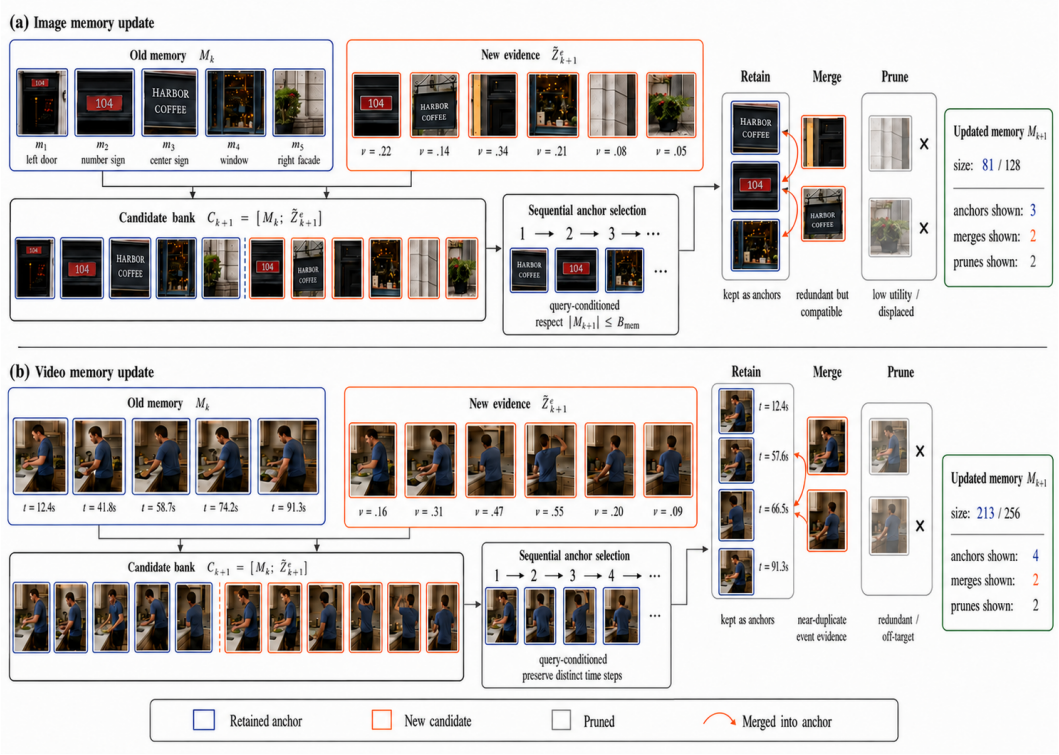


Figure 10: **Inside a WATCH Retain update.** We magnify one memory update from the image trajectory (**top**) and one from the video trajectory (**bottom**). Existing memory M_k and newly acquired evidence $\widetilde{Z}^e_{k+1}$ form the shared candidate bank $C_{k+1} = [M_k; \widetilde{Z}^e_{k+1}]$. Numbers below newly acquired candidates denote their memory-relative novelty ν with respect to M_k. The autoregressive memory policy sequentially selects query-conditioned anchors under B_{mem}; unselected candidates are merged only when they are both semantically redundant and geometrically compatible with a retained anchor, and are otherwise pruned. The image example emphasizes spatially redundant crops at different fidelities, whereas the video example shows that visually similar frames from distinct times can remain separate when they provide complementary temporal evidence. Only representative candidates and decisions are visualized; the reported $|M_{k+1}|$ values refer to the complete persistent memory state.

long sequential-traversal tasks remain difficult. These cases often require high-level reasoning to be interleaved with many successive visual operations, while the current WATCH architecture deliberately keeps the expensive answering backbone outside the perception loop. The result therefore exposes a meaningful limitation: bounded active perception does not by itself solve fully interleaved visual reasoning.

Table 27: **OOD evaluation on ActiveVision.** Frontier-MLLM and human results are taken from the benchmark's tool-free reasoning track; all other rows use our Qwen2.5-VL evaluation protocol.

Method	Accuracy (%)
Preview Only	1.2
Full static input	2.4
Uniform Budget	3.5
WATCH	**9.4**
Best reported tool-free MLLM	10.6
Human	96.1