VISION AS *Efficient* INFORMATION GATHERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern vision-language models perceive every pixel *uniformly* across the frame(s), spending compute on regions irrelevant to the task. While dynamic-resolution and adaptive-tokenization methods adapt the token budget to the input, they remain task-agnostic and must still process the full image at least once. We introduce ACTIVELOOK, which casts visual perception as a decision-making problem. Starting from a coarse glance, we learn a policy that actively decides, turn by turn, *where* to look (bounding box, page index, temporal window) and *how much* to look (scale, frame rate), until it has gathered enough evidence to answer, optimizing accuracy per FLOP. The obstacle is supervision: ground-truth look sequences do not exist. We bootstrap this from a frozen teacher's rollouts, keeping trajectories that verifiably lead to the correct answer, and augment them with synthetic trajectories derived from ground-truth annotations. We distill these trajectories with KL-anchored supervised fine-tuning, followed by reinforcement learning with a *compute reward* that favors efficient looks across the action space: number of looks, bounding box, scale, temporal window, and frame rate. We gate the reward on correctness and schedule the gate to avoid sparse rewards early in training. ACTIVELOOK learns a single policy spanning images, multi-page documents, and videos, that advances the accuracy–FLOPs Pareto frontier across all regimes: matching the base model's accuracy with $7.2\times$ less compute on images and long documents, and $3.5\times$ less on video. At matched input budgets, ACTIVELOOK is 7.9 points more accurate on average while using 27% less compute.

1 INTRODUCTION

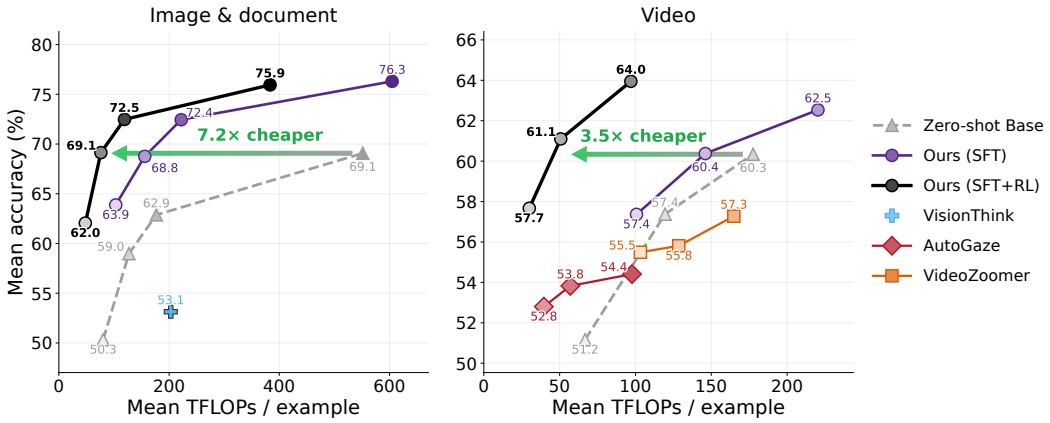


Figure 1: **ACTIVELOOK improves the accuracy–compute frontier.** Our policy matches the base model's accuracy with **7.2**$\times$ less compute on images and **3.5**$\times$ less on video, while outperforming prior efficient-perception methods at every compute budget.

How should a vision-language model (VLM) look at an image? Consider an image of a dog in a park. Asked, *"is there a dog in this image?"* processing the image at high resolution is a pure waste; the answer is obvious from a thumbnail. However, asked, *"what does the tag on the dog's collar say?"* the opposite is true: a coarse tokenization may simply not contain enough detail to recover

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



Figure 2: **How does ACTIVELOOK adapt to different tasks?** Given a task prompt and a coarse visual glance, ACTIVELOOK iteratively decides *where* to look (bounding box, page index, time window) at *how much detail* (scale, frame rate). *(Top)* Given a low-resolution image, ACTIVELOOK produces tight crops and upscales them when needed, especially for small text. *(Middle)*: Given downscaled multi-page documents, it selectively retrieves relevant regions without processing every page at high resolution. *(Bottom)* Given a sparsely sampled video, it retrieves a relevant segment at higher frame rate, revealing evidence not visible in the initial glance.

the answer, no matter how capable the model is. Current VLMs offer one answer to both questions: look everywhere, uniformly, every time. Every region receives the same resolution, every video the same sampling rate, regardless of what the task asks or how much compute it warrants. This ad-hoc strategy blatantly wastes compute on uninformative regions while utterly failing to resolve intricate details where it matters. We argue this is fundamentally wrong, and that vision should instead be understood as an efficient information-gathering process: a sequential decision-making problem in which the model jointly optimizes *where to look* and *how many bits to spend*, given the task.

The central question is: *what is the optimal visual policy, and can it be learned?* Prior work has approached this through hand-designed heuristics of crop sequences or region proposals guided by human labels (Wu & Xie, 2024; Shao et al., 2024a; Shen et al., 2025; Li et al., 2025), task-agnostic token reduction that adapts to visual complexity without accounting for the task (Duggal et al., 2025b;a; Ye et al., 2026; Shi et al., 2026), and adaptive spatial or temporal processing for images and videos (Yang et al., 2025; Ding et al., 2026) that ignore the cost of multi-step perception. All these approaches typically constrain adaptation to a pre-defined strategy or a single axis of visual computation, limiting their ability to allocate compute according to what the task actually requires.

We introduce ACTIVELOOK, a framework that casts visual perception as an *action* the model actively controls. We revisit the classical view of perception as an active, goal-directed process (Bajcsy, 1988; Aloimonos et al., 1988) in the setting of modern VLMs. From an initial coarse glance, the model iteratively decides *where* to look (a spatial bounding box or temporal window) and *how much* to look (spatial scale or frame rate) before halting with an answer. The key challenge is supervision: ground-truth look trajectories do not exist. We bootstrap successful look trajectories from a frozen teacher and augment them with diverse synthetic trajectories derived from ground-truth annotations. We first distill these behaviors through supervised fine-tuning, followed by reinforcement learning using a compute reward that favors efficient looks while preserving accuracy. Accuracy gating is scheduled during training to avoid sparse rewards early while preventing the policy from learning to be cheaply wrong.

ACTIVELOOK Pareto-dominates the accuracy-compute frontier across six benchmarks spanning GUI grounding, visual search, document understanding, and short- and long-form video QA. It matches the base model's accuracy with $7.2\times$ less compute on images and $3.5\times$ less on video. As we scale the visual input, it achieves $+7.9$ accuracy gains on average while still using 27% less compute, with largest gains coming from GUI grounding and document QA ($+24.4$ and $+14.8$ accuracy points). Importantly, our learned policy adapts its computation to the task and visual input: it spends more compute on harder questions and increasingly answers directly from the initial glance as the visual input becomes sufficient. Our contributions are as follows:

- **Vision as efficient sequential information gathering.** We reframe visual perception in VLMs as a sequential decision-making problem, where the model jointly optimizes *where* and *how much* to look conditioned on the task—opening a new and largely under-explored axis of optimization for vision-language models.
- **Compute reward induces adaptive visual policies.** Our accuracy-gated compute reward teaches the policy to reduce unnecessary looks while preserving useful ones, adapting computation to task difficulty and visual complexity beyond what is learned from SFT alone.
- **Single unified adaptive policy across images, documents, and videos.** A single ACTIVELOOK policy learns to adapt spatial and temporal perception across all three regimes, improving the accuracy–compute frontier across six benchmarks spanning visual search, GUI grounding, document understanding, and video QA.

2 RELATED WORK

Adaptive visual computation. A large body of work reduces VLM inference cost by compressing visual inputs or intermediate tokens. Token pruning and merging methods remove redundant visual tokens using similarity or attention, while dynamic-resolution VLMs vary token count with input resolution (Wang et al., 2024; Chen et al., 2024). Recent methods make this allocation increasingly adaptive. For video, AutoGaze (Shi et al., 2026) selects multi-scale spatial and temporal patches before the vision encoder, based on visual content alone without adapting to the task. Closest to our setting are methods that learn, with RL, whether to acquire more detail from a downscaled visual input: VisionThink (Yang et al., 2025) learns whether to request a full-resolution image, while VideoZoomer (Ding et al., 2026) requests higher-frame-rate video segments. These methods adapt *how much* visual information is processed, but each along a single axis. In contrast, ACTIVELOOK jointly learns *where* and *how much* to look in both space and time, conditioned on the task, with a single policy spanning images, documents, and videos.

Active perception and visual search. Prior work enables VLMs to search images through hand-designed procedures: LLM-guided search with a visual working memory in V* (Wu & Xie, 2024), tree search over a zoom hierarchy in ZoomEye (Shen et al., 2025), Monte Carlo tree search with a visual expert in DyFo (Li et al., 2025), or by supervising regions from human box annotations, as in Visual CoT (Shao et al., 2024a). These methods show that looking closer helps, but the search is prescribed rather than learned, and none accounts for the compute each step spends. More recent agentic VLMs learn to acquire visual evidence through cropping, zooming, and tool use (Zheng et al., 2026; Hong et al., 2026; Bai et al., 2025; OpenAI, 2025; Google, 2026; Kimi Team, 2026). These methods learn to acquire additional visual evidence, but optimize primarily for task performance rather than perception efficiency. Our work instead treats visual perception itself as the decision problem: ACTIVELOOK learns where to attend at what detail, with the explicit objective of maximizing accuracy while conserving FLOPs.

3 PROBLEM FORMULATION AND SETUP

Vision as active decision making. We cast visual perception as a sequential decision-making problem, where looking becomes an *action* the model actively controls rather than passively receives. Given a question q and visual input V (an image, multi-page document, or video), the policy π_θ initially observes a coarse *glance* o_0: a downscaled image or a sparsely sampled video. At each turn t, conditioned on the task q and interaction history $H_t = (o_0, a_0, \ldots, o_t)$, the policy decides between two actions: *look* or *halt*. A look specifies *where* to attend (bounding box, page index,

temporal window) and *how much* to look (scale, frame rate) from the original source V, producing a new observation o_{t+1} that is appended to H_t. Importantly, note that the policy can choose to query parts of the image or video that were not initially provided to it, gradually gathering information as necessary. Upon gathering sufficient information, the model produces a "halt" action along with the answer y and terminates the trajectory $\tau = (o_0, a_0, o_1, \ldots, o_T, a_T)$.

Objective: efficient information gathering. Our goal is to maximize accuracy while minimizing compute. We measure the cost of a trajectory by its total forward FLOPs, $\Phi(\tau) = \sum_t \Phi_t$, where Φ_t is the cost at turn t. We evaluate policies along the accuracy–compute frontier: the best achievable accuracy at a given compute budget.

	Where to attend	**How much detail**
Spatial	$\mathbf{b} = (x_1, y_1, x_2, y_2)$: bounding box $\mathcal{I} \subseteq \{1, \ldots, P\}$: page(s) to fetch	$\sigma \in (0, 1]$: resolution scale
Temporal	$\tau^{\text{start}}, \tau^{\text{end}} \in [0, 1]$: time window	$f \in (0, 1]$: frame rate fraction

Look action space. A look re-renders a region of the full-resolution source V, so it recovers detail that is difficult to perceive from the glance o_0. Each look controls *where* to attend and *how much detail* to acquire: a spatial look sets a bounding box $\mathbf{b}$, page indices $\mathcal{I} \subseteq \{1, \ldots, P\}$ for documents) and resolution scale σ; a temporal look adds a time window $(\tau^{\text{start}}, \tau^{\text{end}})$ and a frame-rate fraction f. Video looks may set spatial and temporal fields in the same call, allowing the policy to narrow both *when* and *where* to perceive.

4 METHOD

Learning the policy requires supervision over how to look, but ground-truth look trajectories do not exist. We therefore train ACTIVELOOK in three stages. First, we bootstrap diverse look trajectories from frozen teacher rollouts and task annotations (§4.1). We then distill these trajectories with supervised fine-tuning (SFT), teaching the model the new action space and how to use them effectively (§4.2). Finally, we optimize the policy with reinforcement learning (RL) using an accuracy-gated compute reward, learning when additional visual evidence is worth its compute (§4.3).

4.1 GENERATING AND BOOTSTRAPPING LOOK TRAJECTORIES

Teacher rollouts. For each prompt and initial glance, we sample K candidate trajectories from frozen VLM teachers equipped with the same look action space. We use `GPT-5.5` (OpenAI, 2026), equipped with `LocateAnything-3B` (Wang et al., 2026) for sharper bounding boxes in selected tasks. We de-duplicate trajectories, score their final answers for correctness, and retain successful trajectories as demonstrations of how to look.

Annotation-derived trajectories. For tasks that provide ground-truth annotations (e.g., a bounding box, an evidence page), we construct trajectories directly from this supervision. GUI grounding provides target bounding boxes; document datasets provide evidence pages, on which we additionally localize the relevant region; and for short videos we use temporal segments from VideoZoomer trajectories (Ding et al., 2026), and annotate a bounding box on the video segment's middle frame so the policy learns where to look both temporally and spatially. For long videos, we construct examples with known answer locations from FineVideo (Farré et al., 2024), using its scene timestamps and descriptions as temporal ground truth. We use 2–10 min videos and concatenate same-category clips to create longer videos with realistic distractors. A text-only teacher generates questions from target-scene descriptions, which we verify using only frames within its ground-truth window. We then construct trajectories that look from a sparse-frame glance into this window.

Trajectory augmentation. We further diversify the training trajectories with two synthetic augmentations: repairs and compressions. *Repairs* turn incorrect trajectories into recovery demonstrations by splicing their prefixes with correct trajectories `incorrect[:k] + correct` (A–B). We randomize the splice point k and source correct trajectory to diversify recoveries. We additionally

repair rollouts with valid looks but incorrect answers (C–D). *Compressions* transform correct trajectories into cheaper variants by batching sequential crops into multi-crops in one turn (E) or removing crops that do not contain the answer (F).

4.2 Cold-Start Supervised Fine-tuning

We first fine-tune the policy on the bootstrapped trajectories, teaching it the action space and diverse strategies for acquiring visual evidence. To preserve the base model's general VQA capabilities, we anchor the policy to the frozen pre-trained model π_{ref} with a per-token KL penalty computed over model-generated tokens only (environment observations are masked):

$$\mathcal{L}_{\text{SFT}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{KL}} D_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}})\,. \tag{1}$$

We use $\lambda_{\text{KL}} = 0.1$. SFT thus provides RL with a policy capable of producing diverse, effective spatial and temporal looks. However, it does not explicitly optimize whether an additional look is worth its compute; we address this next with RL.

4.3 Compute-Aware Reinforcement Learning

The reward function shapes the model's action policy: *how* to look and *when* to halt. For each prompt q, GRPO (Shao et al., 2024b) samples trajectories $\mathcal{G}(q) = \{\tau_1, \ldots, \tau_n\}$, each scored with

$$R(\tau) \;=\; w_{\text{acc}} R_{\text{acc}} \;+\; w_{\text{fmt}} R_{\text{fmt}} \;+\; w_{\text{cmp}} R_{\text{cmp}}, \tag{2}$$

where R_{acc} scores the final answer, R_{fmt} checks that each turn emits well-formed actions, and R_{cmp} encourages efficient information gathering.

Compute reward: reward efficient looks. We introduce a compute reward favoring efficient trajectories. This covers the whole look action space—the number of looks, scale, bounding box, time window, and frame rate—shaping not only *whether* to look, but *how much*.

We price a trajectory by the total forward FLOPs it consumes, $\Phi(\tau) = \sum_t \Phi_t$ (§3): the term charges both *how many* looks are taken and *how expensive* each one is. Total cost cannot be compared against a fixed budget: a glance over a 20-page document costs orders of magnitude more than one over a screenshot, so an absolute threshold would grade the input rather than the policy. We therefore define the *relative* compute cost $\rho(\tau)$ by normalizing against the FLOPs of a single-turn pass that answers directly from the glance o_0:

$$\rho(\tau) = \frac{\Phi(\tau)}{\Phi_{\text{single}}(o_0, q)}, \qquad R_{\text{cmp}}(\tau) = \text{clip}\Big(1 - \frac{\rho(\tau) - \rho_{\text{lo}}}{\rho_{\text{hi}} - \rho_{\text{lo}}},\, 0,\, 1\Big), \tag{3}$$

where ρ_{lo}=3.5 and ρ_{hi}=12: a single-turn trajectory scores 1, one that spends four or more passes' worth of compute scores 0, with linear interpolation in between.

Compute reward scheduling. Gating R_{cmp} on correctness from the start makes the reward too sparse when most rollouts are incorrect, while applying it unconditionally rewards efficiency before the policy is accurate. We therefore introduce a gate $\alpha(t) \in [0, 1]$ that is zero for the first fraction d of training steps and ramps linearly to 1 over the next fraction r:

$$\widehat{R}_{\text{cmp}} = R_{\text{cmp}}\big(1 - \alpha(t) + \alpha(t) R_{\text{acc}}\big). \tag{4}$$

Early in training, efficiency is rewarded regardless of correctness; later, only correct and efficient trajectories are rewarded, so the policy is not rewarded by being cheaply wrong.

A trajectory that answers without looking costs a single pass over the glance ($\rho \approx 1$), so it always earns the full compute reward. Rewarding it early, however, can collapse a group to no-look before accuracy becomes informative. We therefore withhold compute reward from no-look trajectories on the same schedule, $\widehat{R}_{\text{cmp}} \leftarrow \alpha(t)\widehat{R}_{\text{cmp}}$. This encourages looking early while allowing no-look trajectories to compete once accuracy becomes informative.

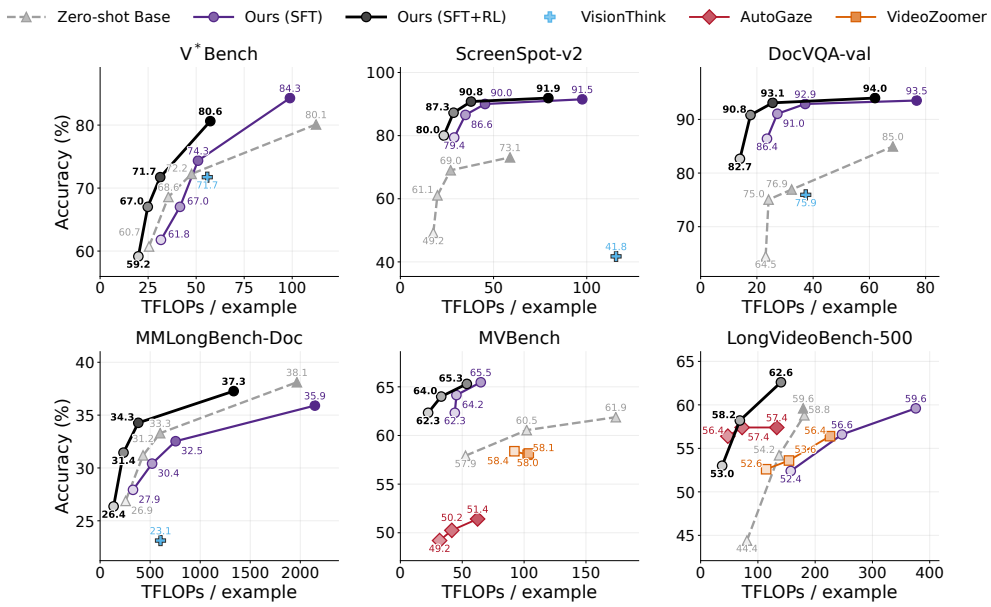


Figure 3: **Accuracy–FLOPs Pareto frontier across six benchmarks.** Each curve varies the initial glance scale $s \in \{0.3, 0.45, 0.6, 1.0\}$ for image and document benchmarks, and glance frame budget (8/16/32 frames) for video; lighter markers denote cheaper settings. ACTIVELOOK consistently dominates the accuracy–FLOPs frontier (upper-left is better).

5 EXPERIMENTS

5.1 SETUP

Models. We use `Qwen3-VL-4B-Instruct` as the policy. For data generation, we use `GPT-5.5` to generate teacher rollouts, verify whether each look contains the answer, and generate bounding-box annotations for selected tasks. We use `gpt-5.4-nano` as an LLM judge to score open-ended answers, both to verify teacher rollouts and compute the RL reward .

Training. Both stages full fine-tune `Qwen3-VL-4B-Instruct` with FSDP on 4×H200 GPUs. For SFT, we train for one epoch on 47,632 trajectories with effective batch size 64, learning rate 10^{-5} with cosine decay and 5% warmup, and a maximum sequence length of 16,384. We use a per-token KL anchor to the base model (λ=0.1). For RL, we run GRPO for one epoch on 12,396 prompts, sampling n=8 rollouts for each of 32 prompts per step. We use learning rate 10^{-6}, KL coefficient 10^{-3} for the SFT initialization, and at most 8 agent turns. Reward weights are w_{acc}=1, w_{fmt}=0.1, and w_{cmp}=0.1; ρ_{lo}=3.5, ρ_{hi}=12, and c=3. The compute-reward schedule uses d=0.05 and r=0.15. All trajectories within a group share the same initial scale or frames.

Evaluation. To compare accuracy–compute trade-offs, we sweep the initial scale over $\{0.3, 0.45, 0.6, 1.0\}$ for all methods and compare their resulting frontiers. The scale sets the glance resolution as an *area* fraction, where an image is rendered at σ_0^2 of its native pixel count. For video, we instead sweep the frame budget over $8, 16, 32$ frames at native resolution.

Baselines. We compare against the following baselines:

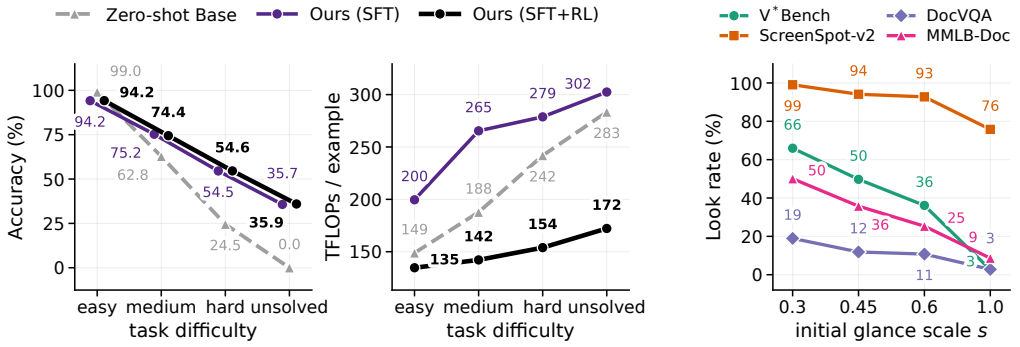
- **Zero-shot base.** The `Qwen3-VL-4B-Instruct` base model prompted with the same look actions and the agentic loop.
- **AutoGaze.** A task-agnostic adaptive token reduction method that selects multi-scale spatial and temporal regions based on visual content. We use its `NVILA-8B-HD-Video` backbone with its recommended gazing settings.

- **VisionThink.** A task-conditioned adaptive resolution method that starts from a downsampled ($2\times$) image and learns whether to request the full-resolution image. It makes a single global choice between two resolutions. We use its default protocol for initial glance.
- **VideoZoomer.** A task-conditioned temporal zooming agent that first views the video at a low frame rate and can request a chosen time segment at a higher frame rate.

Benchmarks. We evaluate on six held-out benchmarks spanning high-resolution images and screenshots, multi-page documents, and videos. For images, *V*Bench* (Wu & Xie, 2024) tests visual search for small targets in high-resolution scenes, and *ScreenSpot-v2* (Wu et al., 2025) tests GUI grounding of small elements in high-resolution screenshots. For documents, *DocVQA-val* (Mathew et al., 2021) asks questions about fine text in single-page document images, and *MMLongBench-Doc* (Ma et al., 2024) requires locating and reasoning over evidence across long documents averaging 47.5 pages. For video, *MVBench* (Li et al., 2024) evaluates temporal understanding across 20 tasks, and *LongVideoBench* (Wu et al., 2024) tests long-vdeo understanding; we use a 500-question subset of its validation set balanced across five duration bands.

5.2 Main result: the accuracy–FLOPs frontier

Figure 3 compares accuracy against forward FLOPs across six benchmarks spanning high-resolution images, screenshots, multi-page documents, and videos. ActiveLook consistently improves the accuracy–FLOPs frontier. Compared with the multi-turn base, it achieves higher accuracy at lower compute on DocVQA (+9.0–18.2 points at 0.60–0.91× the compute), MVBench (+3.4–4.4 points at 0.31–0.43×), and LongVideoBench (+3.8–8.6 points at 0.46–0.77×), while matching its accuracy on V*Bench and MMLongBench-Doc at roughly half the compute. On ScreenSpot-v2, ActiveLook spends 1.3–1.4× more compute to gain 18.8–30.8 accuracy points. Our cheapest setting also outperforms VisionThink on three image and document benchmarks. On video, ActiveLook outperforms VideoZoomer at every matched frame budget while using 1.6–4.1× less compute. It also exceeds AutoGaze on MVBench by 13 points at lower cost (62.3 at 22 vs. 49.2 at 32 TFLOPs); on LongVideoBench it beats AutoGaze from 16 frames (58.2 vs. 57.4) at similar compute. Figure 5 further shows that this efficiency arises from diverse look strategies, including targeted and multi-region search, recovery from imperfect looks, and adaptive spatial and temporal resolution.



(a) Adapt to task difficulty (b) Adapt to visual complexity

Figure 4: **ActiveLook adapts to the task and visual input.** **(a)** RL preserves SFT accuracy while allocating more compute to harder questions (135 → 172 TFLOPs), remaining below the base at every difficulty level. **(b)** As glance resolution increases (s=0.3 → 1.0), ActiveLook increasingly answers directly from the glance, avoiding unnecessary looks.

5.3 Does ActiveLook adapt to the task and visual input?

Adapt to Task Difficulty. We measure task difficulty by the number of glance scales $s \in \{0.3, 0.45, 0.6, 1.0\}$ at which the base model answers each question correctly, from easy (4/4) to unsolved (0/4). Figure 4a averages accuracy and compute across the four image benchmarks and glance scales. SFT substantially improves accuracy on hard and unsolved questions (+30.0 and +35.7 points), but spends more compute than the base across all difficulty levels. RL preserves SFT

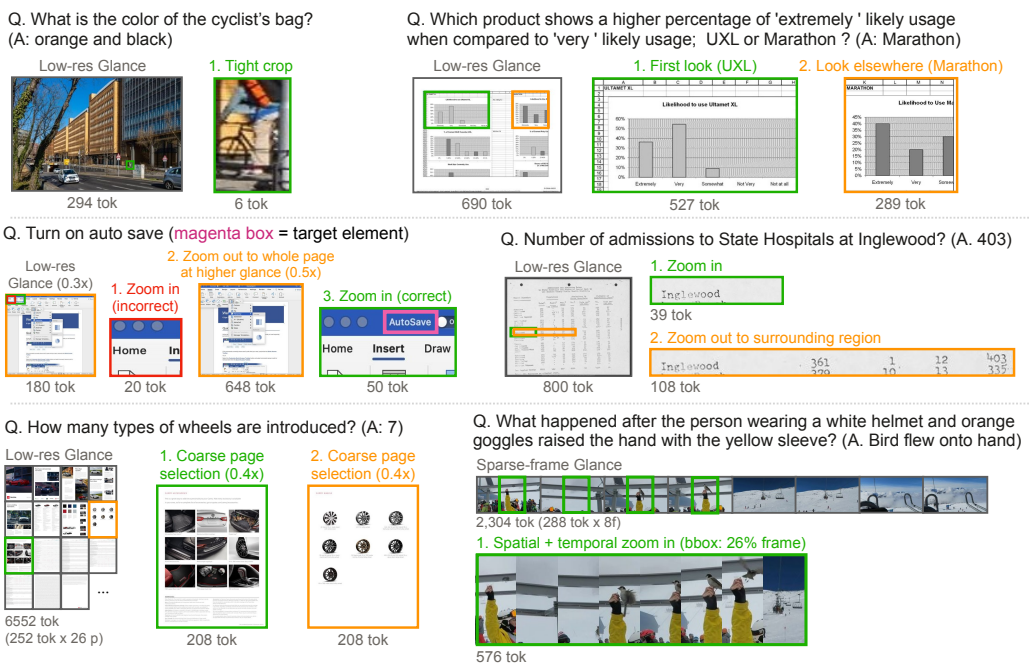


Figure 5: **ACTIVELOOK exhibits diverse look strategies.** *(Top)*: It produces tight crops for visual search, or looks across multiple regions when the task requires integrating spatially distributed evidence. *(Middle)*: It *recovers* from incorrect looks by zooming out to the whole page at higher glance or to the surrounding region. *(Bottom)*: It adapts resolution to the task, viewing coarse pages ($0.4\times$) when sufficient for counting, and jointly zooms in space and time for efficient video understanding.

accuracy (within 0.8 points) while reducing its compute by 33–46%. More importantly, it allocates compute according to difficulty ($135 \rightarrow 172$ TFLOPs), while remaining below the base ($149 \rightarrow 283$ TFLOPs) at every level.

Adaptation to visual complexity. We next vary the information available in the initial visual glance. As its resolution increases from s=0.3 to 1.0, ACTIVELOOK looks less often on every benchmark: from $66 \rightarrow 3\%$ on V*Bench, $50 \rightarrow 9\%$ on MMLongBench-Doc, $19 \rightarrow 3\%$ on DocVQA, and $99 \rightarrow 76\%$ on ScreenSpot-v2 (Figure 4b). The policy thus increasingly answers directly when the glance is sufficient, rather than producing unnecessary looks. In contrast, the base shows little adaptation; its look rate on V*Bench remains at 83–88% across scales. Qualitative examples are in Appendix 8.

5.4 RL ABLATIONS

We ablate each component of RL to isolate its effect on the accuracy–FLOPs frontier, as summarized in Figure 6, with the corresponding training dynamics in Figure 7. Numbers are averaged over held-out V*Bench, ScreenSpot-v2 and MMLongBench-Doc at each input scale s=0.3/0.45/0.6/1.0.

Compute reward improves efficiency; accuracy gate preserves useful looks. Removing the compute reward leaves the policy looking on essentially every example (90% tool use and 2.7 turns on average), spending $2.1\times$ the compute of the full system for only $+0.6$ accuracy points: it has no incentive to stop looking. Removing the accuracy gate causes the opposite failure. With α=0, token reduction is rewarded regardless of correctness, so both accuracy and look rate collapse: over training its look rate decays from 82% to 3% between steps 150–200, and accuracy falls with it (Figure 7). At evaluation, tool use is 0.0% with the trajectory length exactly one turn across all benchmarks and input scales. This hurts low-scale inputs where looking can recover missing detail, reducing 2.1 accuracy points on average over s=0.3–0.6. Thus, the compute reward drives efficiency, while accuracy gating preserves looks that improve accuracy when resolution is scarce.

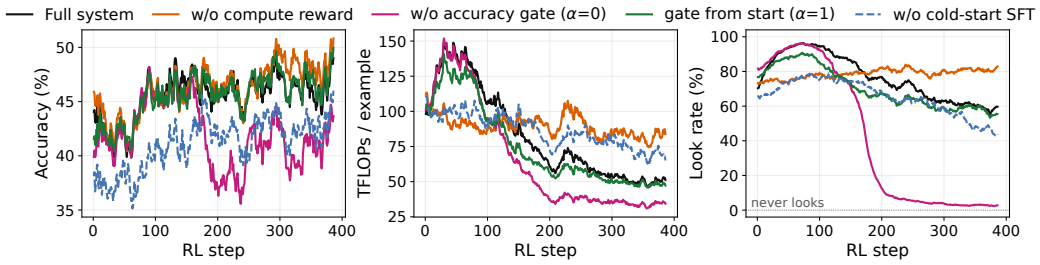


Figure 7: **RL ablation training dynamics.** Training accuracy, TFLOPs/example, and look rate over RL steps (left to right). Without the compute reward, compute remains high and look rate increases. Without accuracy gating (α=0), compute falls sharply but the policy collapses to rarely looking (82% $\rightarrow$ 3%), hurting accuracy. Gating from start (α=1) holds look rate at 55% while reducing average TFLOPs/example (102 $\rightarrow$ 47), preserving useful looks to achieve high accuracy.

Gate scheduling buys accuracy at lower scales. Delaying and scheduling the accuracy gate (full system), rather than applying it at full strength from step 0 (α=1), improves accuracy by 0.7 points on average over s=0.3–0.6 at 1.14$\times$ the compute. The gain is concentrated on V*Bench, where small targets often require additional looks to resolve: scheduling improves accuracy by 2.1 and 2.6 points at s=0.45 and s=0.6, respectively, at 1.21$\times$ the compute, while ScreenSpot-v2 and MMLongBench-Doc remain within 0.6 points. At s=1.0, the two schedules converge, with gating from the start marginally ahead. These results suggest that the delayed gate encourages early exploration: tool use remains $\sim$11 points higher over steps 100–200 (Figure 7), allowing the policy to discover useful look behaviors that immediate gating suppresses.

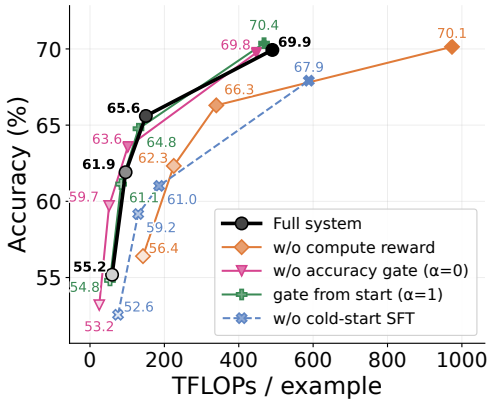


Figure 6: **RL ablations.** The accuracy gate improves accuracy at low-resolution glance (s=0.3–0.6). Without the compute reward, the policy uses 2.1$\times$ more compute for little accuracy gain. Without cold-start SFT, it scores lower and costs more at every scale.

Cold-start SFT teaches look quality. Without cold-start SFT, held-out accuracy rises from 42% to 57% over training, but remains below the SFT-initialized models, which start at 52%–56% and finish at 57%–60% (Figure 7). More importantly, RL from scratch is worse on both axes of the accuracy–FLOPs frontier: despite far fewer looks (21% vs 53% look rate), it spends 1.23$\times$ the compute while scoring 3.0 accuracy points lower, and trails at every input scale ($-$2.0 to $-$4.6). The deficit is largest when looks must be placed precisely: $-$5.4 points on ScreenSpot-v2 and $-$3.0 on V*Bench, against $-$0.6 on MMLongBench-Doc. RL alone learns to look less, but not to look well.

6 Conclusion

We introduced ActiveLook, which casts visual perception as efficient information gathering: a policy learns to adapt *where* and *how much* to look according to the task and visual input. We make this behavior learnable by bootstrapping diverse look trajectories for supervised fine-tuning and subsequently optimizing the policy with a compute-aware reinforcement learning objective. A single policy learns adaptive spatial and temporal perception across images, multi-page documents, and videos, substantially improving the accuracy–compute frontier. Beyond reducing FLOPs, active perception can also improve accuracy, suggesting that selectively acquiring task-relevant evidence can be more effective than uniformly processing all available visual information. More broadly, our results suggest that visual computation need not be fixed at inference time: models can learn to allocate perception adaptively, spending computation only where it is useful.

AI USE STATEMENT

Tasks with required disclosure. We used generative AI tools to generate synthetic look trajectories. Specifically, we used frozen VLMs, including `GPT-5.5` to generate candidate look trajectories. We also used generative AI to construct annotation-derived training examples. These uses and the associated filtering procedures are described in the paper. We have not used generative AI tools to formulate mathematical claims or proofs, interpret experimental results, or determine the conclusions of this work.

Other tasks. We used generative AI tools to assist with debugging, visualization, and suggestions for improving the clarity and concision of the manuscript. All AI-assisted code, visualizations, experimental outputs, and manuscript edits were reviewed by the authors. Experimental results and claims were verified by the authors. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REFERENCES

John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 1(4):333–356, 1988. doi: 10.1007/BF00133571.

Shuai Bai et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988. doi: 10.1109/5.5968.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. How far are we to GPT-4V? Closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67(12):220101, 2024.

Yang Ding, Xin Lai, Yizhen Zhang, Wei Li, Ruihang Chu, and Yujiu Yang. VideoZoomer: Reinforcement-learned temporal focusing for long video reasoning. In *International Conference on Learning Representations (ICLR)*, 2026.

Shivam Duggal, Sanghyun Byun, William T Freeman, Antonio Torralba, and Phillip Isola. Single-pass adaptive image tokenization for minimum program search. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Shivam Duggal, Phillip Isola, Antonio Torralba, and William T Freeman. Adaptive length image tokenization via recurrent allocation. In *International Conference on Learning Representations (ICLR)*, 2025b.

Miquel Farré, Andi Marafioti, Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. Finevideo. `https://huggingface.co/datasets/HuggingFaceFV/finevideo`, 2024.

Google. Introducing agentic vision in Gemini 3 Flash. *https://blog.google/innovation-and-ai/technology/developers-tools/agentic-vision-gemini-3-flash/*, 2026.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. Deepeyesv2: Toward agentic multimodal model. In *International Conference on Learning Representations (ICLR)*, 2026.

Kimi Team. Kimi K2.5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Geng Li et al. DyFo: A training-free dynamic focus visual search for enhancing LMMs in fine-grained visual understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9098–9108, 2025.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. MVBench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22195–22206, 2024.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yu-Gang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. MMLongBench-Doc: Benchmarking long-context document understanding with visualizations. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 37, pp. 95963–96010, 2024.

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)*, 2021.

OpenAI. Thinking with images. `https://openai.com/index/thinking-with-images/`, 2025.

OpenAI. Introducing GPT-5.5. `https://openai.com/index/introducing-gpt-5-5/`, 2026.

Hao Shao et al. Visual CoT: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024a.

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

Haozhan Shen et al. ZoomEye: Enhancing multimodal LLMs with human-like zooming capabilities through tree-based image exploration. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6613–6629, 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, 2026.

Peng Wang et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Shihao Wang, Shilong Liu, Yuanguo Kuang, Xinyu Wei, Yangzhou Liu, Zhiqi Li, Yunze Man, Guo Chen, Andrew Tao, Guilin Liu, et al. LocateAnything: Fast and high-quality vision-language grounding with parallel box decoding. In *European Conference on Computer Vision*, pp. 336–357. Springer, 2026.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 37, pp. 28828–28857, 2024.

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

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and Yu Qiao. OS-ATLAS: Foundation action model for generalist GUI agents. In *International Conference on Learning Representations (ICLR)*, 2025.

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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Haotian Ye, Qiyuan He, Jiaqi Han, Puheng Li, Jiaojiao Fan, Zekun Hao, Fitsum Reda, Yogesh Balaji, Huayu Chen, Sheng Liu, et al. InfoTok: Adaptive discrete video tokenizer via information-theoretic compression. In *International Conference on Learning Representations (ICLR)*, 2026.

Ziwei Zheng et al. DeepEyes: Incentivizing "thinking with images" via reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026.

A Appendix

A.1 Dataset statistics and details

Table 1: **Cold-start SFT dataset by source**.

Dataset	Task	Trajectory source	Glance	Rows
MME-RealWorld (OCR)	high-res VQA	GPT-5.5 + LocateAny	0.3	2,963
MME-RealWorld (visual search)	high-res VQA	GPT-5.5 + LocateAny	0.3	2,984
MME-RealWorld (OCR, anchor)	high-res VQA	base answer, no look	0.3	1,449
GroundCUA	GUI (desktop)	rule (element box)	0.30/.45/.60	2,966
OS-Atlas-mobile	GUI (mobile)	rule (element box)	0.30/.45/.60	3,093
OS-Atlas-web	GUI (web)	rule (element box)	0.30/.45/.60	2,941
MP-DocVQA	multi-page doc (med. 4 pp.)	GT page + GPT-5.5	0.3	13,803
SlideVQA	slide deck (med. 20 pp.)	GT page + GPT-5.5	0.3	1,133
VideoZoomer	long-video VQA	VideoZoomer + GPT-5.5	$\leq$64f	4,397
FineVideo	video search / needle	constructed	32f/64f	4,387
Total				**47,632**

Table 2: **Synthetic data generation strategies.** Repair operations (A–D) recycle incorrect trajectories. Trajectory augmentations (E–G) compress correct trajectories into cheaper variants.

Strategy	Condition	Operation
Repairs		
A (prefix append)	wrong look	`incorrect[:k]`+correct
B (overlap append)	wrong look, shared opening	`incorrect[:k]+correct[d:]`
C (correct answer)	right looks, incorrect answer	copy a correct rollout's final turn
D (reformat answer)	right looks, mis-formatted answer	rewrite `<answer>`
Compressions		
E (multicrop collapse)	correct, sequential crops	batch into one multi-region look
F (prune-to-answerable)	correct, redundant crops	keep only answerable crops

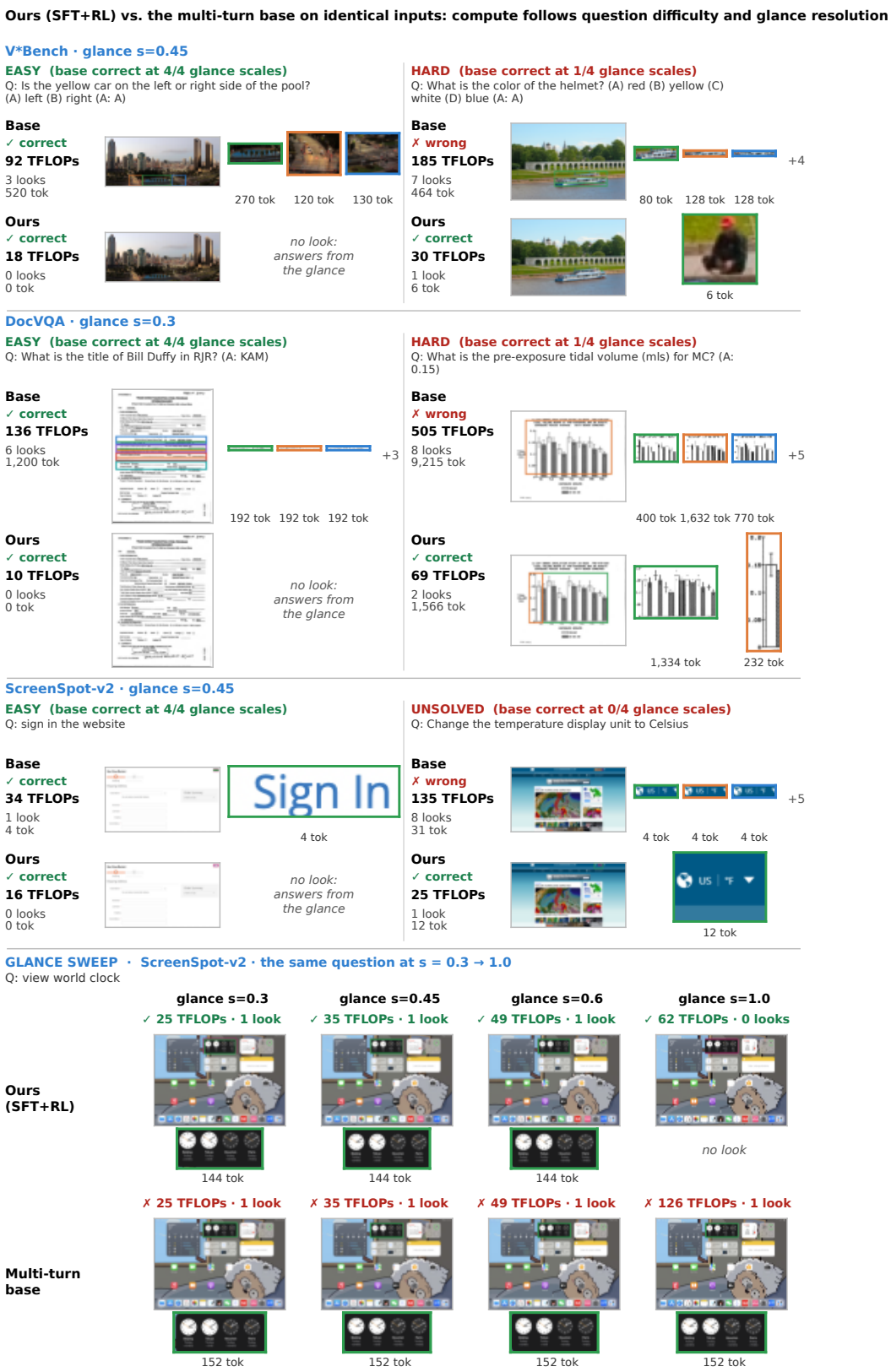


Figure 8: **Comparison to base model on task and visual adaptivity.**