LATENT LOOKAHEAD: RECOVERING ACTION-CONDITIONED PREDICTIVE STRUCTURE FROM FROZEN VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained vision–language models (VLMs) are increasingly reused as internal representations for multimodal agents, yet it remains unclear whether states optimized for understanding preserve information about how visual evidence changes after an action. We ask whether question- and history-conditioned representations from a frozen VLM contain recoverable, local action-conditioned predictive structure. Latent Lookahead uses a shallow readout and a low-capacity transition model to predict future latent states, then ranks candidate actions through short latent rollouts while leaving the backbone fixed. Across three VLMs and image, document, and video decision environments, learned transitions outperform persistence and action-independent predictors. The advantage persists over short open-loop rollouts, but weakens with horizon, policy-induced distribution shift, and poorer training coverage. From identical decision states, predicted futures increase best-realized-action match from 55.9% to 64.3% and reduce action regret from 0.0283 to 0.0146 relative to direct scoring. Latent Lookahead also improves task performance by about two points on average with roughly 30% additional compute and no extra backbone calls per real interaction step. These results show that local predictive information can be recovered from frozen VLM representations and used for short-horizon visual evidence selection.

1 INTRODUCTION

Visual reasoning often requires deciding what to observe before deciding what to answer. An agent may need to inspect another image region, move to a different document page, or revisit a video segment before it has enough evidence to resolve a query. Vision–language models (VLMs) provide rich internal representations of the current observation together with the language query and interaction history, and these representations are increasingly reused as the state of multimodal agents. The information sufficient for answering from the evidence currently available may not preserve the distinctions needed to anticipate which observation will be useful next. This matters when visual access is costly and the agent cannot inspect every available observation. We therefore ask whether information about the consequences of alternative visual actions remains recoverable from the internal states of a frozen VLM.

Predictive state modeling has been studied under several representation-learning regimes. Dreamer and TD-MPC2 learn state representations jointly with dynamics for model-based control (Hafner et al., 2025; Hansen et al., 2024). I-JEPA and V-JEPA learn visual features through prediction in representation space (Assran et al., 2023; Bardes et al., 2024). DINO-WM further shows that action-conditioned dynamics can be learned over frozen pretrained visual features and used for goal-directed planning (Zhou et al., 2025a). We study a different representation regime: the hidden state of a VLM after visual evidence has been integrated with a language query and interaction history. Whether such task-conditioned multimodal states retain enough information to anticipate subsequent visual observations remains unclear.

Our hypothesis is that this information can be recovered locally with limited additional capacity. Let z_t denote a compact readout of the frozen VLM representation at interaction step t. If the representation retains action-dependent information about future evidence, a small transition model

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

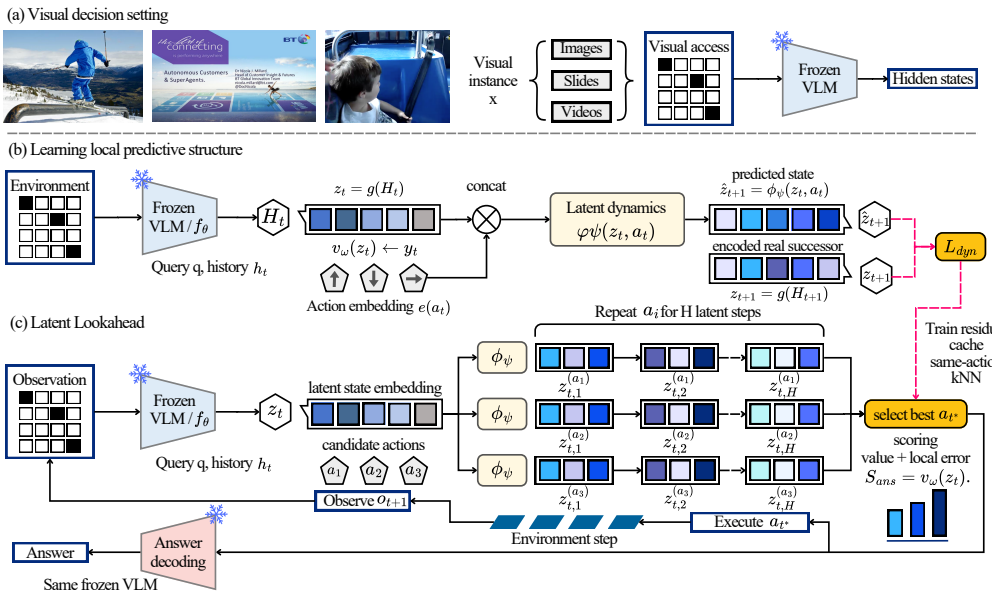

Figure 1: Overview of Latent Lookahead. A shallow readout exposes a compact VLM state, a lightweight dynamics model learns action-conditioned transitions from real successors, and short latent rollouts evaluate candidate visual actions before execution, while the frozen backbone remains unchanged throughout.

should predict the state reached after executing an action. We focus on short-horizon behavior, where predicted changes may remain informative before accumulated transition error and distribution shift dominate. We evaluate this property both as a prediction problem and through the quality of actions selected from the predicted futures.

We introduce *Latent Lookahead* to operationalize this hypothesis. A shallow learned readout maps frozen VLM hidden states to a compact latent representation. A small action-conditioned transition model is trained from offline interaction trajectories, while a value head estimates discounted task success. At inference time, candidate navigation actions are propagated through short, non-branching latent rollouts and scored using predicted values together with a local transition-error estimate. Only the selected action is executed in the environment, and newly observed evidence is encoded at the next real interaction step. Simulated futures therefore require no additional VLM evaluations. Figure 1 summarizes the method.

Across three frozen VLM backbones and image, document, and video decision environments, learned action-conditioned transitions predict successor states more accurately than persistence and action-independent alternatives. Their advantage remains over short open-loop rollouts and weakens with rollout length, policy-induced distribution shift, and weaker local training support. To connect this predictive signal to action quality, we evaluate competing choices from identical decision origins and execute each candidate under the same continuation policy. On Qwen3-VL-8B-Instruct, using predicted futures increases pooled agreement with the empirically best next action from 55.9% to 64.3% and reduces mean action regret from 0.0283 to 0.0146 relative to a separately optimized direct action scorer. This isolates an improvement at the level of the next visual decision, before subsequent policy behavior can introduce additional differences.

Using these local predictions for action selection improves downstream task performance by about two points on average across the evaluated environments, with roughly 30% additional computation and no extra backbone calls per real interaction step. The experiments identify a recoverable predictive property of task-conditioned VLM representations: action-dependent information remains accessible through a shallow readout, supports short-range state prediction, and improves the selection of visual evidence. This property degrades with longer imagined trajectories and weaker state–action coverage. Our conclusions are limited to discrete visual-access decisions and do not establish Markov sufficiency, long-horizon world modeling, or continuous physical control.

2 Related Work

Predictive representations and latent world models. Latent world models support planning by predicting state evolution in a learned representation space. Dreamer and TD-MPC2 learn the state representation jointly with the transition model for model-based control (Hafner et al., 2025; Hansen et al., 2024). Predictive representation methods instead shape the feature space through prediction itself: I-JEPA predicts target image representations from context, while V-JEPA extends feature prediction to video (Assran et al., 2023; Bardes et al., 2024). Related work has explored contrastive predictive objectives, image world models, and multimodal predictive representations (Mo & Tong, 2024; Garrido et al., 2024; Chen et al., 2025; Assran et al., 2025; Sun et al., 2026). DINO-WM is particularly close to our setting: it learns action-conditioned dynamics over frozen pretrained DINO features and uses them for goal-directed planning (Zhou et al., 2025a); DINO-world further studies future prediction in pretrained DINO feature spaces (Baldassarre et al., 2025). These results establish that frozen visual features can support latent prediction. We study a different representation object: a VLM hidden state that has already integrated visual evidence with a language query and interaction history, and ask whether local action-dependent information remains recoverable from this task-conditioned state.

Multimodal agents and active visual access. Vision–language models are widely used in agents that combine perception, reasoning, and action. MM-REACT composes language models with specialist visual tools (Yang et al., 2023), while ViperGPT generates programs that invoke visual modules for structured reasoning (Surís et al., 2023). VisualWebArena and VisualAgentBench extend this setting to web, embodied, and interface-based decision tasks (Koh et al., 2024; Liu et al., 2025). Other systems explicitly control what visual evidence is acquired: training-free visual search and active-perception methods adaptively inspect image regions or observations before answering (Zhou et al., 2025b; Zhang et al., 2025a), while UNeMo incorporates predicted future visual states into navigation decisions (Huang et al., 2026). FOREWARN similarly combines latent dynamics with VLM-based evaluation for robot policy steering (Wu et al., 2025). These systems show that multimodal representations can support sequential evidence acquisition and control. Our focus is narrower: we measure whether the frozen VLM state itself retains recoverable information about how alternative visual-access actions change the subsequent internal state, and whether that information improves the next observation choice.

Test-time reasoning and latent intervention. Test-time computation can improve model behavior through repeated decoding, search, or direct manipulation of internal representations. Chain-of-thought prompting, self-consistency, and Tree-of-Thoughts increase reasoning depth through additional generation or explicit search (Wei et al., 2022; Wang et al., 2023; Yao et al., 2023). A separate line modifies latent representations during inference. CREST intervenes on identified attention heads (Zhang et al., 2025b), ATLAS adapts latent steering using an external verifier (Nguyen & Le, 2026), and spectrum-aware steering operates in learned subspaces of vision–language representations (Dafnis & Metaxas, 2025). Test-time prototype shifting likewise adapts class representations while keeping the encoders frozen (Sui et al., 2025). Latent Lookahead uses the frozen representation differently: it learns a low-capacity transition model over that state and evaluates actions through predicted state evolution. This lets us separate direct action scoring from the additional information provided by short-horizon prediction.

3 Problem Setup

We consider finite-horizon visual decision problems in which an agent receives a visual instance x and a language query q, while only part of the available visual information is exposed at each step. An access configuration c_t determines the current observation,

$$o_t = \mathcal{O}(x, c_t).$$

Conditioned on o_t, q, and the interaction history h_t, the agent selects an action $a_t \in \mathcal{A}$. Navigation updates the access configuration as

$$c_{t+1} = \mathcal{T}(c_t, a_t),$$

revealing another observation from the same underlying instance, while ANSWER terminates the episode. The horizon is bounded by T. This formulation covers spatial, document, and temporal visual access; environment-specific actions, horizons, and metrics are described in Section 5.

Let f_θ be a pretrained VLM whose parameters remain fixed. Given the current observation, query, and history, it produces multimodal hidden states

$$H_t = f_\theta(o_t, q, h_t) \in \mathbb{R}^{L_t \times D}.$$

A shallow readout g maps these features to a compact representation

$$z_t = g(H_t) \in \mathbb{R}^d.$$

We do not assume that z_t is Markov-sufficient or fully describes the underlying environment. Our question is whether this compact readout retains information about how the VLM state changes under alternative visual-access actions. The implementation of g is given in Section 4.

If such information is present, a low-capacity transition model should predict the successor representation,

$$\hat{z}_{t+1} = \phi(z_t, a_t) \approx z_{t+1}.$$

We keep ϕ deliberately small to limit the task structure introduced outside the frozen representation. This hypothesis leads to three empirical questions: whether prediction depends on the executed action beyond state persistence, whether the recovered transition structure remains informative under short compositions, and whether it retains predictive signal under moderate policy-induced shifts in state visitation. We refer to these properties collectively as local action-conditioned predictive structure. Latent Lookahead, introduced next, provides a lightweight mechanism for recovering this structure and using predicted futures to select visual observations.

4 LATENT LOOKAHEAD

Latent Lookahead learns local dynamics over the compact states defined in Section 3 and uses predicted states to select visual observations. Offline trajectories provide supervision for transition prediction and task value estimation. At inference, candidate actions are evaluated through latent rollouts, without querying the VLM for imagined observations.

4.1 COMPACT STATE READOUT

The readout aggregates the top-layer hidden states $H_t \in \mathbb{R}^{L_t \times D}$ by averaging over the token sequence, then projects the result to a d-dimensional state:

$$z_t = g(H_t) = \mathrm{norm}\left(W\ \mathrm{mean}(H_t) + b\right), \tag{1}$$

where L_t is the sequence length, D is the backbone hidden dimension, $W \in \mathbb{R}^{d \times D}$ and $b \in \mathbb{R}^d$ are learned parameters, and norm denotes L_2 normalization. The same readout encodes current and real successor observations and provides the inputs to both dynamics and value prediction. Additional representation learning is restricted to this linear projection, while the VLM parameters remain fixed.

4.2 LEARNING LOCAL DYNAMICS AND STATE VALUE

We collect offline trajectories with the frozen VLM policy using only training instances. The transition dataset $\mathcal{D}_{\mathrm{dyn}}$ contains triples (z_t, a_t, z_{t+1}) for executed navigation actions with a successor observation. Both states are encoded using the same backbone and readout. Completed episodes provide a binary success indicator $r \in \{0, 1\}$ for value supervision.

Action-conditioned dynamics. A two-layer MLP ϕ_ψ predicts the next state from z_t and a learned action embedding $e(a_t) \in \mathbb{R}^{d_a}$, where d_a is the embedding dimension. Let σ denote the nonlinear activation and $[\cdot\|\cdot]$ denote concatenation. The model parameters ψ comprise the action embeddings, weights, and biases. Prediction and its training objective are

$$\begin{aligned} \hat{z}_{t+1} &= \phi_\psi(z_t, a_t) \\ &= \mathrm{norm}(W_2 \sigma(W_1[z_t \| e(a_t)] + b_1) + b_2), \\ \mathcal{L}_{\mathrm{dyn}} &= \mathbb{E}_{(z_t, a_t, z_{t+1}) \sim \mathcal{D}_{\mathrm{dyn}}} \left[\|\phi_\psi(z_t, a_t) - z_{t+1}\|_2^2 \right]. \end{aligned} \tag{2}$$

The target is the normalized representation obtained after executing the real action. This objective measures one-step prediction error in the readout space, with limited model capacity for learning the transition.

State value. A value head $v_\omega(z_t) \in [0, 1]$ estimates how useful a state is for completing the task. Let τ denote the episode's terminal step and $\gamma \in (0, 1)$ the discount factor. Each visited state is paired with the target $y_t = r\gamma^{\tau-t}$, forming $\mathcal{D}_{\text{val}} = \{(z_t, y_t)\}$. Successful trajectories receive targets that increase toward termination, while unsuccessful trajectories receive zero targets. We fit the value head using binary cross-entropy:

$$\mathcal{L}_{\text{val}} = \mathbb{E}_{(z_t,y_t)\sim\mathcal{D}_{\text{val}}} \left[-y_t \log v_\omega(z_t) - (1 - y_t) \log(1 - v_\omega(z_t))\right]. \tag{3}$$

The readout, action embeddings, dynamics model, and value head are optimized jointly using these objectives. The value estimate represents discounted success under the trajectory-collection distribution and serves as a score when evaluating imagined states.

4.3 Local Error-Aware Latent Lookahead

Imagined trajectories can reach states poorly represented in the offline data. We use a local prediction-error estimate to penalize reliance on uncertain transitions. For cached training transition i, let e_i denote its squared prediction residual. For a query (z, a), $\mathcal{N}_k(z, a)$ indexes the k nearest cached input states under cosine distance, restricted to transitions with action a. The residual and local estimate are

$$\begin{aligned} e_i &= \|\phi_\psi(z_i, a_i) - z_{i+1}\|_2^2, \\ \epsilon(z, a) &= \frac{1}{k} \sum_{i\in\mathcal{N}_k(z,a)} e_i. \end{aligned} \tag{4}$$

We use ϵ as an empirical error proxy without assuming probabilistic calibration.

At real interaction step t, the frozen VLM encodes the observed context once to obtain z_t. The candidate set $\mathcal{C}_t$ contains the valid navigation actions and ANSWER. For each navigation action a, we simulate H transitions:

$$z_{t,0}^{(a)} = z_t, \qquad z_{t,h+1}^{(a)} = \phi_\psi\left(z_{t,h}^{(a)}, a\right), \quad h = 0, \ldots, H - 1. \tag{5}$$

Because ϕ_ψ outputs normalized states, real and imagined states remain in the same normalized readout space. Each rollout repeats the same action, with dynamics parameters shared across candidates and steps. For a fixed candidate set, the number of simulated transitions grows linearly with H.

Navigation scores average the predicted state values and subtract the local error penalty, weighted by $\lambda \geq 0$. The terminal action uses the current-state value:

$$\begin{aligned} S_{\text{nav}}(a) &= \frac{1}{H} \sum_{h=1}^{H} \left[v_\omega\left(z_{t,h}^{(a)}\right) - \lambda\, \epsilon\left(z_{t,h-1}^{(a)}, a\right)\right], \\ S_{\text{ans}} &= v_\omega(z_t). \end{aligned} \tag{6}$$

The error term is evaluated at the source state of the transition that produces $z_{t,h}^{(a)}$. For ANSWER, S_{ans} is an operational stopping score under the same discounted-success target, without an imagined transition.

Let $S(a) = S_{\text{ans}}$ for ANSWER and $S(a) = S_{\text{nav}}(a)$ for navigation. The agent selects $a_t^\star = \arg\max_{a\in\mathcal{C}_t} S(a)$. A selected navigation action is executed once, and planning restarts from the state encoded after the new observation. Selecting ANSWER produces the final response through the original VLM decoder and terminates the episode. All simulated transitions use the latent model without additional backbone evaluations. Model sizes, optimization settings, cache construction, and the defaults for H, k, and λ are specified in Section 5.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We evaluate Latent Lookahead on three finite-horizon visual decision environments: GQA-PatchEnv for spatial image inspection (Hudson & Manning, 2019), SlideVQA-Env for slide navigation (Tanaka et al., 2023), and NExT-QA-Env for temporal video exploration (Xiao et al., 2021). Each environment preserves the original benchmark instances, official splits, and scoring rules while adding a discrete visual-access interface with maximum horizons of 8, 10, and 12 steps. We report GQA balanced accuracy, SlideVQA EM/F1 and joint EM/F1, and NExT-QA accuracy. Experiments use three frozen backbones: Qwen3-VL-8B-Instruct, Phi-4-multimodal-instruct, and Gemma-3n-E4B-it. For each backbone, we collect offline trajectories from the official training splits using its frozen reactive policy and train one latent module jointly across the three environments. The default configuration uses $d = 128$, rollout horizon $H = 2$, $k = 10$ neighbors for local error estimation, and reliability weight $\lambda = 0.1$. For predictive diagnostics, the trained readout is frozen so that all predictors share the same latent space and held-out transitions. We compare learned dynamics with persistence, random, action-independent, and linear action-conditioned predictors using cosine similarity to the encoded real successor. Multi-step prediction is evaluated open-loop from matched starting states and legal action sequences, and we additionally test planner-induced transitions and prediction error under varying local training coverage.

Downstream comparisons include the reactive policy, one-step lookahead, a direct action scorer, and Latent Lookahead with $H = 2$, together with three- and five-trajectory reactive policies as compute references. The direct scorer learns its own readout and scoring head from the same frozen inputs and offline data under a comparable optimization budget. To isolate the immediate observation choice, we conduct a same-origin diagnostic with Qwen3-VL-8B-Instruct at held-out states containing at least two legal navigation actions. Competing methods receive the same observation, query, history, access configuration, and remaining budget; each candidate action is executed from this common origin, after which all branches use the same frozen reactive continuation policy with matched decoding seeds. We report realized utility, agreement with the empirically best action, and empirical regret. All methods use the same evaluation instances and decoding settings: temperature 0.2, top-p 0.95, at most 32 generated tokens, and greedy scoring for multiple-choice answers. Downstream results are averaged over three decoding seeds, while same-origin comparisons use paired differences with bootstrap resampling grouped by the underlying image, slide deck, or video. Relative FLOPs are normalized to the complete single-trajectory reactive query, and latency is reported as median complete-query runtime on a single A100 with batch size one. Additional environment, optimization, trajectory, seed, and profiling details are provided in the appendix.

5.2 CHARACTERIZING PREDICTIVE STRUCTURE

We first test whether a compact readout of the frozen VLM state retains information about action-dependent state changes. Table 1 compares one-step prediction in a shared, fixed readout space. Across all three backbones and environments, action-conditioned predictors improve over persistence and action-independent prediction, and the two-layer dynamics model further improves over the linear alternative. For Qwen3-VL-8B, learned dynamics reach cosine similarities of 0.889, 0.871, and 0.844 on GQA, SlideVQA, and NExT-QA. The same ordering holds for Phi-4-mm and Gemma-3n-E4B, although the margins vary across environments. We do not compare absolute cosine values across backbones as measures of representation quality, since each backbone induces a different latent space.

We next test whether this predictive signal remains informative when transitions are composed. Figure 2a evaluates open-loop prediction with Qwen3-VL-8B-Instruct for up to four latent steps. Fidelity decreases with horizon, but learned dynamics remain closer to the encoded real states than persistence throughout the evaluated range. On GQA, cosine similarity decreases from 0.889 at one step to 0.682 at four steps, compared with 0.827 to 0.612 for persistence. SlideVQA and NExT-QA show the same ordering, with faster degradation on the temporal video environment. The result supports a short-horizon interpretation of the recovered dynamics: action-conditioned prediction remains informative under limited composition, while its reliability decreases as rollout depth grows.

Table 1: One-step predictive fidelity in the fixed readout space. Values are cosine similarity to encoded real successors. (mean ± std. over three latent-model runs)

Backbone	Environment	Persistence	Action-indep.	Linear	Learned
Qwen3-VL-8B	GQA	0.827±0.004	0.857±0.004	0.873±0.003	**0.889±0.003**
	SlideVQA	0.791±0.005	0.826±0.001	0.847±0.007	**0.871±0.003**
	NExT-QA	0.758±0.006	0.799±0.005	0.818±0.004	**0.844±0.004**
Phi-4-mm	GQA	0.821±0.005	0.845±0.003	0.860±0.008	**0.873±0.003**
	SlideVQA	0.802±0.004	0.840±0.003	0.858±0.010	**0.876±0.003**
	NExT-QA	0.747±0.005	0.788±0.005	0.805±0.004	**0.827±0.005**
Gemma-3n-E4B	GQA	0.807±0.005	0.833±0.005	0.844±0.004	**0.853±0.004**
	SlideVQA	0.782±0.006	0.819±0.004	0.830±0.009	**0.845±0.004**
	NExT-QA	0.744±0.003	0.781±0.006	0.795±0.006	**0.808±0.005**

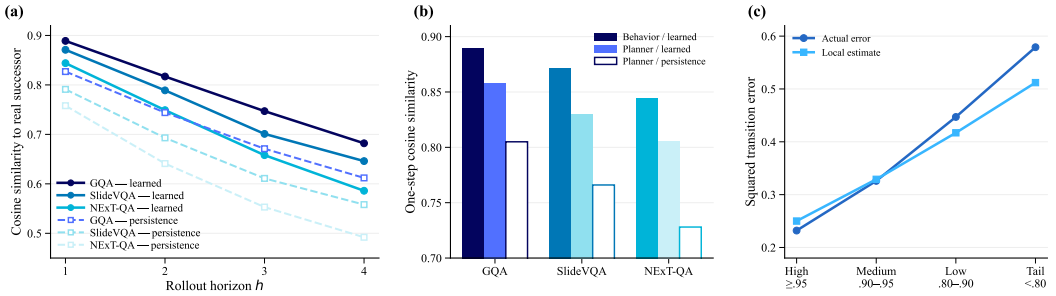

Figure 2: Characterizing local predictive structure with Qwen3-VL-8B-Instruct. (a) Open-loop prediction across rollout horizons. (b) One-step fidelity under behavior-policy and planner-induced states. (c) Actual and estimated transition error as local training coverage decreases.

Figure 2b–c further shows where this structure becomes less reliable. Under planner-induced state visitation, learned one-step fidelity falls from 0.889 to 0.858 on GQA, from 0.871 to 0.830 on SlideVQA, and from 0.844 to 0.805 on NExT-QA, while remaining above the corresponding persistence baseline. Prediction error also increases as local training support weakens: mean squared error rises from 0.232 in the highest-coverage bucket to 0.579 in the tail. The residual-based estimate follows the same trend but underestimates the tail error, reaching 0.512. We therefore use it as an empirical reliability signal rather than a calibrated uncertainty estimate. Overall, the recovered structure is local in three senses: it is action-dependent, remains informative over short compositions and moderate policy shift, and deteriorates as state–action coverage weakens.

5.3 Predictive Structure for Decision Making

The previous section establishes that action-conditioned state evolution is predictable over short horizons. We next ask whether this information improves the visual decisions made by the agent. Table 2a compares Latent Lookahead with controls that remove or shorten the predictive component while keeping the frozen Qwen3-VL-8B-Instruct backbone. A direct action scorer already improves substantially over the reactive policy, reaching 64.10, 75.90, and 83.85 on GQA, SlideVQA, and NExT-QA. Predicting future states provides an additional gain: Latent Lookahead reaches 64.90, 76.70, and 84.30, respectively. The margin over direct scoring is modest but consistent across the three environments, ranging from 0.45 to 0.80 points. One-step lookahead and action-independent dynamics recover smaller gains, suggesting that both action conditioning and short-horizon state evolution contribute to the final decision rule.

The full-policy comparison does not by itself isolate whether predicted futures improve the immediate observation choice, since subsequent actions may also differ. We therefore evaluate both methods from identical decision origins, execute each candidate navigation action, and then return every branch to the same frozen reactive continuation policy. Table 2b reports the pooled results. Latent Lookahead increases agreement with the empirically best realized action from 55.9% to 64.3%, while mean empirical regret falls from 0.0283 to 0.0146. Realized utility rises from 0.752 to 0.765,

Table 2: Decision-making analysis with Qwen3-VL-8B-Instruct. (a) Downstream controls isolate the contribution of short-horizon transition prediction. (b) Same-origin evaluation compares actions under an identical continuation policy.

(a) Downstream mechanism controls

Method	GQA Acc.	SlideVQA J-F1	NExT-QA Acc.
Reactive	62.70±0.24	73.60±0.27	82.70±0.18
One-step (H=1)	63.00±0.21	75.30±0.25	83.50±0.20
Action-independent dynamics	63.51±0.26	75.42±0.23	83.29±0.19
Direct scorer	64.10±0.23	75.90±0.22	83.85±0.17
Latent Lookahead (H=2)	**64.90±0.19**	**76.70±0.21**	**84.30±0.16**

(b) Same-origin action quality

Metric	Direct scorer	Latent Lookahead	Difference
Realized utility ↑	0.752	**0.765**	+0.0138
Best-action match (%) ↑	55.9	**64.3**	+8.4 pp
Empirical regret ↓	0.0283	**0.0146**	−0.0137

The pooled paired difference in realized utility has a 95% confidence interval of $[+0.005, +0.022]$.

Table 3: Downstream performance across frozen VLM backbones. We report GQA balanced accuracy, SlideVQA joint F1, and NExT-QA accuracy (mean ± std. over three decoding seeds).

Backbone	Method	GQA	SlideVQA	NExT-QA
Qwen3-VL-8B	Reactive	62.70±0.24	73.60±0.27	82.70±0.18
	Latent Lookahead	**64.90±0.19**	**76.70±0.21**	**84.30±0.16**
Phi-4-mm	Reactive	61.60±0.28	72.30±0.25	81.90±0.20
	Latent Lookahead	**63.30±0.22**	**74.60±0.24**	**83.30±0.18**
Gemma-3n-E4B	Reactive	60.10±0.31	71.20±0.28	80.80±0.22
	Latent Lookahead	**61.60±0.27**	**73.10±0.25**	**82.10±0.21**

with a pooled paired difference of $+0.0138$ and a 95% confidence interval of $[+0.005, +0.022]$. These results connect future-state prediction to better next-action selection under a controlled continuation policy, rather than relying only on final performance differences between complete policies.

We next test whether this downstream benefit persists across model families. Table 3 reports results for three frozen VLM backbones. Latent Lookahead improves every backbone–environment pair. Qwen3-VL-8B gains $+2.2$ points on GQA, $+3.1$ joint-F1 points on SlideVQA, and $+1.6$ points on NExT-QA. Phi-4-mm gains 1.7, 2.3, and 1.4 points, while Gemma-3n-E4B gains 1.5, 1.9, and 1.3 points. Averaged over the nine backbone–environment pairs, the improvement is 1.89 points in the corresponding task metric.

The magnitude of the downstream gain varies across backbones and environments even though its direction is consistent. This complements the predictive analysis in Section 5.2: stronger one-step fidelity does not translate into a fixed amount of task improvement, and the backbone with the highest predictive fidelity is not uniformly the one with the largest downstream gain. Recoverable state dynamics therefore provide useful information for selecting visual observations, while the resulting benefit also depends on the task, backbone, and how predicted states are converted into action values.

5.4 Efficiency and Scope

Latent Lookahead trades a small amount of latent computation for fewer repeated VLM evaluations. Figure 3a compares this trade-off with multi-sample reactive inference. With $H = 2$, Latent Lookahead uses $1.30\times$ the FLOPs of a single reactive trajectory and improves the mean task metric by 2.30 points across GQA, SlideVQA, and NExT-QA. Three- and five-trajectory reactive inference increase the cost to $3.0\times$ and $5.0\times$, while yielding average gains of 0.67 and 1.20 points. We treat this as a compute reference rather than a universal efficiency ranking, since multi-sample inference repeats

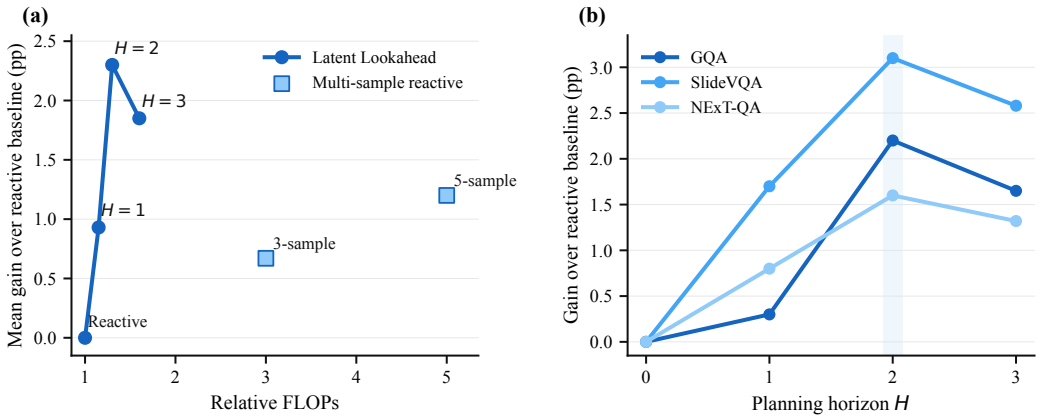

Figure 3: Efficiency and scope with Qwen3-VL-8B-Instruct. (a) Mean task gain over the reactive policy as a function of total query FLOPs; multi-sample decoding is shown as a compute reference. (b) Downstream gains across planning horizons.

backbone computation whereas Latent Lookahead keeps imagined transitions inside the lightweight latent model.

The useful rollout depth is limited. Figure 3b shows that all three environments improve from $H = 1$ to $H = 2$, after which performance declines. The gains on GQA are $+0.30$, $+2.20$, and $+1.65$ points for $H = 1, 2, 3$; SlideVQA gives $+1.70$, $+3.10$, and $+2.58$; NExT-QA gives $+0.80$, $+1.60$, and $+1.32$. This decline coincides with the loss of open-loop predictive fidelity in Figure 2a, identifying a common short-horizon operating regime in which additional lookahead remains useful before prediction quality deteriorates.

The current formulation is therefore limited to discrete visual-access actions and short, non-branching rollouts that repeatedly apply one candidate action. Its computational advantage comes from keeping these rollouts in latent space rather than increasing the number of backbone evaluations. Longer horizons, mixed-action branching, and continuous physical control introduce additional model and search error beyond the present setup. We use $H = 2$ as the default operating point for the evaluated environments, not as a general optimum.

6 Discussion and Conclusion

Our results suggest that task-conditioned VLM representations retain more than the information required to answer from the current observation. Through a shallow readout, they expose local action-dependent structure that can anticipate how the internal state changes after additional visual access. This structure is useful for decision making: predicted futures improve next-action selection beyond direct scoring alone. Predictive fidelity and downstream benefit, however, do not vary in lockstep across backbones, tasks, or rollout horizons, indicating that accurate latent prediction is only one component of effective visual evidence selection. The experiments also delineate the scope of the current approach. Prediction becomes less reliable as rollout depth increases, as the induced policy departs from the trajectory-collection distribution, and as local training coverage weakens. Latent Lookahead therefore targets short-horizon decisions over discrete visual-access actions rather than long-horizon simulation or continuous physical control. Because the predictive state is obtained through a learned readout, our results establish recoverable predictive structure in frozen VLM representations without implying that the original hidden states are Markov-sufficient. The local error estimate provides a useful reliability signal, but it is not probabilistically calibrated and becomes less accurate in poorly covered regions.

Latent Lookahead provides a lightweight way to recover and use this predictive structure without adapting the VLM backbone. More broadly, the results indicate that representations trained for multimodal understanding can preserve information about the consequences of future observations even without an explicit predictive objective. Extending this property to richer action spaces, longer horizons, and settings where actions alter the physical world remains an open direction.

AI USE STATEMENT

Generative AI tools were used during the preparation of this work for several supporting purposes. They assisted with language polishing and rewriting, literature retrieval and related-work discovery, and early-stage research ideation and experimental planning. They were also used to help organize and refine portions of the manuscript, including the presentation of the method, experimental analyses, figures, and tables. All methodological choices, experimental procedures, reported measurements, citations, and scientific claims were independently reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

We document the interactive visual-access environments, frozen-backbone interfaces, latent readout, action-conditioned transition model, value estimation, local error estimation, and latent-planning procedure in the main paper and appendix. The appendix further provides environment parameters, training and decoding settings, predictive controls, same-origin evaluation protocol, statistical procedures, hyperparameter analyses, and computational profiling details. All reported comparisons use fixed evaluation instances and controlled backbone and decoding settings. Code, configurations, and additional reproducibility materials will be released with the supplementary material where permitted by the submission policy.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael G. Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023. doi: 10.1109/CVPR52729.2023.01499. URL https://dblp.org/rec/conf/cvpr/AssranDMBVRLB23.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew J. Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *CoRR*, abs/2506.09985, 2025. doi: 10.48550/ARXIV.2506.09985. URL https://dblp.org/rec/journals/corr/abs-2506-09985.

Federico Baldassarre, Marc Szafraniec, Basile Terver, Vasil Khalidov, Francisco Massa, Yann LeCun, Patrick Labatut, Maximilian Seitzer, and Piotr Bojanowski. Back to the features: DINO as a foundation for video world models. *CoRR*, abs/2507.19468, 2025. doi: 10.48550/ARXIV.2507.19468. URL https://dblp.org/rec/journals/corr/abs-2507-19468.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://dblp.org/rec/journals/tmlr/BardesGPCRLAB24.

Delong Chen, Mustafa Shukor, Théo Moutakanni, Willy Chung, Jade Yu, Tejaswi Kasarla, Allen Bolourchi, Yann LeCun, and Pascale Fung. VL-JEPA: Joint embedding predictive architecture for vision-language. *CoRR*, abs/2512.10942, 2025. doi: 10.48550/ARXIV.2512.10942. URL https://dblp.org/rec/journals/corr/abs-2512-10942.

Konstantinos M. Dafnis and Dimitris N. Metaxas. Test-time spectrum-aware latent steering for zero-shot generalization in vision-language models. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.52202/085713-5056. URL https://dblp.org/rec/conf/nips/DafnisM25.

Quentin Garrido, Mahmoud Assran, Nicolas Ballas, Adrien Bardes, Laurent Najman, and Yann LeCun. Learning and leveraging world models in visual representation learning. *CoRR*,

abs/2403.00504, 2024. doi: 10.48550/ARXIV.2403.00504. URL https://dblp.org/rec/journals/corr/abs-2403-00504.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy P. Lillicrap. Mastering diverse control tasks through world models. *Nat.*, 640(8059):647–653, 2025. doi: 10.1038/S41586-025-08744-2. URL https://dblp.org/rec/journals/nature/HafnerPBL25.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024. URL https://dblp.org/rec/conf/iclr/00010024.

Changxin Huang, Lv Tang, Zhaohuan Zhan, Lisha Yu, Runhao Zeng, Zun Liu, Zhengjie Wang, and Jianqiang Li. UNeMo: Collaborative visual-language reasoning and navigation via a multimodal world model. In *AAAI Conference on Artificial Intelligence*, pp. 18315–18323, 2026. doi: 10.1609/AAAI.V40I22.38895. URL https://dblp.org/rec/conf/aaai/HuangTZYZLWL26.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 6700–6709. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00686. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Hudson_GQA_A_New_Dataset_for_Real-World_Visual_Reasoning_and_Compositional_CVPR_2019_paper.html.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In *Annual Meeting of the Association for Computational Linguistics*, pp. 881–905, 2024. doi: 10.18653/V1/2024.ACL-LONG.50. URL https://dblp.org/rec/conf/acl/KohLJDLHNZSF24.

Xiao Liu, Tianjie Zhang, Yu Gu, Iat Long Iong, Xixuan Song, Yifan Xu, Shudan Zhang, Hanyu Lai, Jiadai Sun, Xinyue Yang, Yu Yang, Zehan Qi, Shuntian Yao, Xueqiao Sun, Siyi Cheng, Qinkai Zheng, Hao Yu, Hanchen Zhang, Wenyi Hong, Ming Ding, Lihang Pan, Xiaotao Gu, Aohan Zeng, Zhengxiao Du, Chan Hee Song, Yu Su, Yuxiao Dong, and Jie Tang. VisualAgentBench: Towards large multimodal models as visual foundation agents. In *International Conference on Learning Representations*, 2025. URL https://dblp.org/rec/conf/iclr/LiuZ0ISXZLSYYQY25.

Shentong Mo and Peter Tong. Connecting joint-embedding predictive architecture with contrastive self-supervised learning. In *Advances in Neural Information Processing Systems*, 2024. URL https://dblp.org/rec/conf/nips/MoT24.

Tuc Nguyen and Thai Le. ATLAS: Adaptive test-time latent steering with external verifiers for enhancing LLMs reasoning. *CoRR*, abs/2601.03093, 2026. doi: 10.48550/ARXIV.2601.03093. URL https://dblp.org/rec/journals/corr/abs-2601-03093.

Elaine Sui, Xiaohan Wang, and Serena Yeung-Levy. Just shift it: Test-time prototype shifting for zero-shot generalization with vision-language models. In *Winter Conference on Applications of Computer Vision*, pp. 825–835, 2025. URL https://dblp.org/rec/conf/wacv/SuiWY25.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: Enhancing vision-language-action model with latent world model. *CoRR*, abs/2602.10098, 2026. doi: 10.48550/ARXIV.2602.10098. URL https://dblp.org/rec/journals/corr/abs-2602-10098.

Dídac Surís, Sachit Menon, and Carl Vondrick. ViperGPT: Visual inference via python execution for reasoning. In *IEEE/CVF International Conference on Computer Vision*, 2023. doi: 10.1109/ICCV51070.2023.01092. URL https://dblp.org/rec/conf/iccv/SurisMV23.

Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. SlideVQA: A dataset for document visual question answering on multiple images. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 13636–13645. AAAI Press, 2023. doi: 10.1609/AAAI.V37I11.26598. URL https://doi.org/10.1609/AAAI.V37I11.26598.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://dblp.org/rec/conf/iclr/0002WSLCNCZ23.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022. URL https://dblp.org/rec/conf/nips/Wei0SBIXCLZ22.

Yilin Wu, Ran Tian, Gokul Swamy, and Andrea Bajcsy. From foresight to forethought: VLM-in-the-loop policy steering via latent alignment. *CoRR*, abs/2502.01828, 2025. doi: 10.48550/ARXIV.2502.01828. URL https://dblp.org/rec/journals/corr/abs-2502-01828.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. NExT-QA: next phase of question-answering to explaining temporal actions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 9777–9786. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00965. URL https://doi.org/10.1109/CVPR46437.2021.00965.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. MM-REACT: Prompting ChatGPT for multimodal reasoning and action. *CoRR*, abs/2303.11381, 2023. doi: 10.48550/ARXIV.2303.11381. URL https://dblp.org/rec/journals/corr/abs-2303-11381.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023. URL https://dblp.org/rec/conf/nips/YaoYZS00N23.

Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. MLLMs know where to look: Training-free perception of small visual details with multimodal LLMs. In *International Conference on Learning Representations*, 2025a. URL https://dblp.org/rec/conf/iclr/0002KCI25.

Zhenyu Zhang, Xiaoxia Wu, Zhongzhu Zhou, Qingyang Wu, Yineng Zhang, Pragaash Ponnusamy, Harikaran Subbaraj, Jue Wang, Shuaiwen Leon Song, and Ben Athiwaratkun. Understanding and steering the cognitive behaviors of reasoning models at test-time. *CoRR*, abs/2512.24574, 2025b. doi: 10.48550/ARXIV.2512.24574. URL https://dblp.org/rec/journals/corr/abs-2512-24574.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning*, pp. 79115–79135, 2025a. URL https://dblp.org/rec/conf/icml/ZhouPLP25.

Yunqi Zhou, Chengjie Jiang, Chun Yuan, and Jing Li. Look where it matters: Training-free ultra-hr remote sensing VQA via adaptive zoom search. *CoRR*, abs/2511.20460, 2025b. doi: 10.48550/ARXIV.2511.20460. URL https://dblp.org/rec/journals/corr/abs-2511-20460.

APPENDIX

A ENVIRONMENT AND IMPLEMENTATION DETAILS

Unless otherwise stated, ablation results are averaged over the same three training or decoding seeds used in the main experiments.

A.1 VISUAL-ACCESS ENVIRONMENTS

We construct three finite-horizon visual-access environments from GQA (Hudson & Manning, 2019), SlideVQA (Tanaka et al., 2023), and NExT-QA (Xiao et al., 2021). The underlying instances, official data splits, and answer scoring are unchanged. The environment wrapper determines which part of the visual instance is exposed at each interaction step and provides a discrete navigation interface before the terminal ANSWER action.

In GQA-PatchEnv, the access configuration determines the currently observed spatial region of an image, with a maximum of 8 interaction steps. SlideVQA-Env exposes one part of a presentation deck at a time and allows navigation within the same deck for up to 10 steps. NExT-QA-Env controls access to temporal regions of a video and uses a maximum horizon of 12. We retain the benchmark metrics and use GQA balanced accuracy, SlideVQA joint F1, and NExT-QA accuracy as the primary downstream measures.

Table A1: Visual-access environments used in the experiments.

Environment	Visual access	Horizon	Primary metric
GQA-PatchEnv	Spatial region	8	Balanced accuracy
SlideVQA-Env	Slide	10	Joint F1
NExT-QA-Env	Temporal region	12	Accuracy

Initialization, legal navigation actions, boundary handling, and observation construction are fixed across compared methods. A method only receives the visual evidence exposed by the environment at the current real interaction step.

A.2 LATENT-MODEL CONFIGURATION

We use Qwen3-VL-8B-Instruct, Phi-4-multimodal-instruct, and Gemma-3n-E4B-it as frozen backbones. The top-layer multimodal hidden sequence is mean pooled and projected to a 128-dimensional state. Both encoded states and transition-model outputs are L_2 normalized. The transition model is a two-layer MLP with a learned 64-dimensional action embedding and hidden width 1024 using GELU. The value head uses hidden width 256 and a scalar sigmoid output.

Because encoded targets and predicted successors have unit norm, the dynamics loss and cosine fidelity satisfy

$$\|\hat{z} - z\|_2^2 = 2 - 2\cos(\hat{z}, z). \tag{7}$$

The squared-error training objective and cosine diagnostic therefore measure the same angular discrepancy up to an affine transformation. Normalization is also applied after every predicted transition, so multi-step rollouts remain in the same normalized state space used to train the value head.

The latent modules are trained with AdamW for 20,000 updates using a batch size of 512. The learning rate is 3×10^{-4} with 1,000 warm-up steps followed by cosine decay to 3×10^{-5}. We use weight decay 10^{-4}, Adam coefficients $(0.9, 0.999)$, and gradient clipping at 1.0. Dynamics and value losses receive equal weight. The success discount is $\gamma = 0.95$. Validation is performed every 500 updates, and the checkpoint with the lowest sum of validation dynamics and value loss is retained.

Backbone forward passes use bfloat16, while latent losses are accumulated in FP32. All backbone parameters remain frozen.

Table A2: Default training and planning configuration.

Setting	Value
Latent dimension d	128
Action embedding	64
Dynamics hidden width	1024
Dynamics activation	GELU
Value hidden width	256
State / prediction normalization	L_2
Optimizer	AdamW
Learning rate	3×10^{-4}
Minimum learning rate	3×10^{-5}
Weight decay	10^{-4}
Adam coefficients	$(0.9, 0.999)$
Batch size	512
Training updates	20,000
Warm-up updates	1,000
Gradient clipping	1.0
Dynamics:value loss weight	$1:1$
Discount γ	0.95
Validation interval	500 updates
Planning horizon H	2
Nearest neighbors k	10
Reliability weight λ	0.1

Table A3: Effect of task conditioning on predictive fidelity with Qwen3-VL-8B-Instruct. Entries are cosine similarity to the encoded real state after one and two open-loop transitions.

Representation	GQA		SlideVQA		NExT-QA	
	$h=1$	$h=2$	$h=1$	$h=2$	$h=1$	$h=2$
Visual only	.846	.758	.823	.731	.781	.673
Visual + query	.876	.798	.854	.764	.812	.713
Visual + query + history	**.889**	**.817**	**.871**	**.789**	**.844**	**.749**

A.3 Offline Trajectory Collection

For each backbone and environment, we collect 20,000 training trajectories and 2,000 held-out validation trajectories using the corresponding frozen reactive policy. The validation trajectories are drawn from data reserved from the official training split and are not used to fit latent-model parameters. Each executed navigation action with a real successor observation contributes a transition (z_t, a_t, z_{t+1}) to $\mathcal{D}_{\text{dyn}}$. Visited states are paired with their discounted terminal-success targets for $\mathcal{D}_{\text{val}}$.

One latent module is trained per backbone using the union of trajectories from the three environments. The residual cache stores prediction errors for training transitions and is queried using cosine distance among transitions with the same action. Unless an ablation states otherwise, the full cache is retained.

B Additional Predictive-Structure Analyses

B.1 Role of Query and Interaction-History Conditioning

The main experiments use VLM states conditioned on the current visual observation, language query, and interaction history. To separate this setting from prediction over generic visual features, we repeat the Qwen3-VL-8B-Instruct analysis while progressively restoring the task-conditioning inputs. The transition architecture and training budget are unchanged.

Visual evidence alone already supports non-trivial prediction. Query conditioning increases fidelity in all three environments, while the interaction history provides a further gain. The history contri-

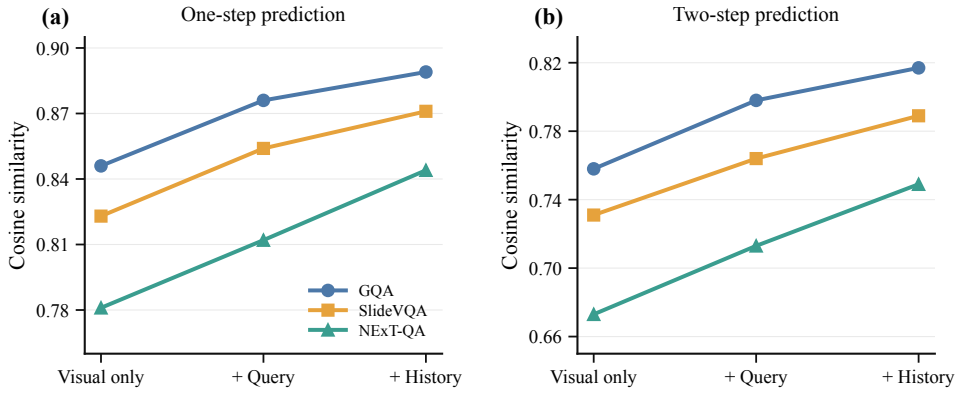

Figure A1: Effect of task conditioning on predictive fidelity with Qwen3-VL-8B-Instruct. (a) One-step and (b) two-step open-loop cosine similarity as query and interaction-history conditioning are added to the visual representation.

Table A4: Additional one-step predictive controls with Qwen3-VL-8B-Instruct. The metric is cosine similarity to the encoded real successor.

Predictor	GQA	SlideVQA	NExT-QA
Random unit state	.057	.043	.061
Persistence	.827	.791	.758
Shuffled-action dynamics	.854	.821	.796
Action-independent	.857	.826	.799
Linear action-conditioned	.873	.847	.818
Learned dynamics	**.889**	**.871**	**.844**

bution is largest on NExT-QA, where the current temporal observation can be ambiguous without information about previously visited regions. The same ordering appears in the pooled same-origin diagnostic: best-realized-action match is 59.8%, 62.2%, and 64.3% for visual-only, visual-plus-query, and full conditioning, respectively. Their realized utilities are 0.756, 0.760, and 0.765.

Figure A1 visualizes the same trend across one- and two-step prediction.

B.2 Additional Predictive Controls

Table A4 adds two controls to the predictors reported in the main paper. The random-state control samples a unit-norm latent without using the current state. The shuffled-action control uses the same transition architecture after permuting action labels in the training transitions. Its performance is close to the action-independent predictor, indicating that a substantial part of the remaining gain depends on the correspondence between action identity and state change.

B.3 Multi-Step Prediction Across Backbones

The main text reports open-loop prediction for Qwen3-VL-8B-Instruct. Table A5 extends the same protocol to all three backbones. Predictions are composed without correction from intermediate real observations, and each target is the state encoded after executing the corresponding action sequence in the environment.

All three backbones retain an advantage over persistence throughout the evaluated horizon, while absolute fidelity declines with composition. The gap is not constant across backbones or environments. Phi-4-mm, for example, has slightly higher one-step fidelity on SlideVQA than on GQA, whereas its longer-horizon prediction degrades at a similar rate to Qwen3-VL-8B.

Table A5: Open-loop predictive fidelity across backbones. Values are cosine similarity to encoded real states.

Backbone	Env.	Predictor	$h = 1$	$h = 2$	$h = 3$	$h = 4$
Qwen3-VL-8B	GQA	Learned	**.889**	**.817**	**.747**	**.682**
		Persistence	.827	.744	.671	.612
	Slide	Learned	**.871**	**.789**	**.701**	**.646**
		Persistence	.791	.693	.611	.558
	NExT	Learned	**.844**	**.749**	**.658**	**.586**
		Persistence	.758	.641	.553	.492
Phi-4-mm	GQA	Learned	**.873**	**.801**	**.731**	**.666**
		Persistence	.821	.735	.660	.601
	Slide	Learned	**.876**	**.797**	**.711**	**.648**
		Persistence	.802	.704	.619	.560
	NExT	Learned	**.827**	**.731**	**.638**	**.563**
		Persistence	.747	.627	.537	.476
Gemma-3n-E4B	GQA	Learned	**.853**	**.775**	**.700**	**.631**
		Persistence	.807	.720	.643	.580
	Slide	Learned	**.845**	**.759**	**.673**	**.607**
		Persistence	.782	.681	.596	.541
	NExT	Learned	**.808**	**.706**	**.611**	**.536**
		Persistence	.744	.619	.529	.466

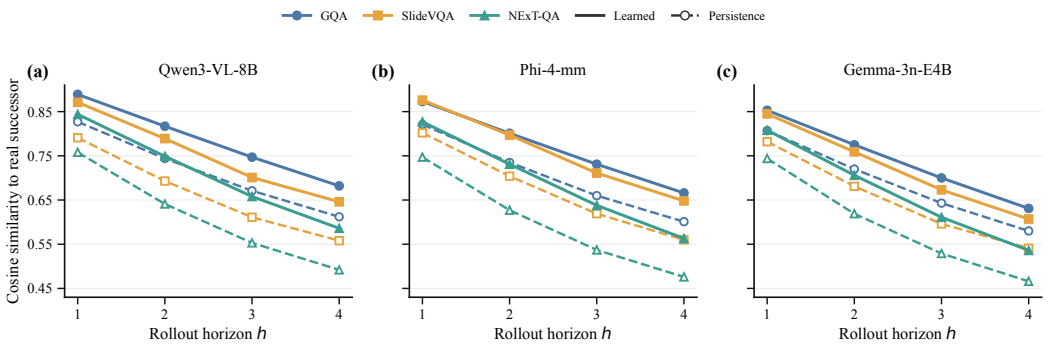

Figure A2: Open-loop predictive fidelity across frozen VLM backbones. Solid lines show learned action-conditioned dynamics and dashed lines show persistence; colors distinguish GQA, SlideVQA, and NExT-QA. All panels use the same vertical scale.

B.4 Policy-Induced Distribution Shift

We evaluate the trained Qwen3-VL-8B predictor on real transitions encountered by the reactive trajectory-collection policy and by Latent Lookahead. Planner-induced transitions are excluded from latent-model training.

Planner-induced visitation reduces learned fidelity by 0.031, 0.041, and 0.039 on GQA, SlideVQA, and NExT-QA. The learned predictor nevertheless remains above persistence on those planner-induced states.

B.5 Coverage and Local Error Estimation

We group held-out transitions by cosine similarity to nearby same-action training states. Prediction error rises as this local support weakens. The residual-based estimate follows the same ordering but underestimates the realized error in the lower-coverage buckets.

The Spearman associations between the estimate and realized error are 0.54, 0.44, and 0.50 for GQA, SlideVQA, and NExT-QA, respectively. We therefore use $\epsilon(z, a)$ as a local ranking signal, without treating it as calibrated uncertainty.

Table A6: One-step fidelity under behavior-policy and planner-induced state visitation.

Environment	Behavior learned	Planner learned	Planner persistence
GQA	.889	**.858**	.805
SlideVQA	.871	**.830**	.766
NExT-QA	.844	**.805**	.728

Table A7: Transition error as a function of local training coverage.

Coverage	Share	Actual error	Local estimate
High ($\geq .95$)	40%	.232	.250
Medium (.90–.95)	30%	.326	.329
Low (.80–.90)	20%	.447	.417
Tail ($< .80$)	10%	.579	.512

C Decision Diagnostics

C.1 Same-Origin Evaluation Protocol

The same-origin evaluation separates the immediate navigation decision from differences in later policy behavior. We sample 600 held-out decision origins from each environment, for 1,800 origins in total. Each origin is non-terminal and contains at least two legal navigation actions. Competing methods receive the same observation, query, interaction history, access configuration, and remaining budget.

For origin j, let $U_j(a)$ denote the realized utility obtained after executing candidate navigation action a and then following the frozen reactive policy using three matched continuation seeds. The empirical best navigation action is

$$a_j^\star \in \arg\max_{a \in \mathcal{A}_j^{\text{nav}}} U_j(a), \tag{8}$$

and the regret of a method selecting $\hat{a}_j$ is

$$R_j = \max_{a \in \mathcal{A}_j^{\text{nav}}} U_j(a) - U_j(\hat{a}_j). \tag{9}$$

If several actions attain the same maximum utility within numerical precision, all are treated as empirically optimal for best-action-match evaluation. Their common maximum is used to compute regret. ANSWER is evaluated separately and is not included in the best-navigation-action comparison.

Candidate branches are generated only for offline evaluation. Their future observations are never provided to the deployed planner.

C.2 Same-Origin Action Quality

Direct scoring and Latent Lookahead choose the same navigation action on 58.7%, 61.9%, and 67.3% of GQA, SlideVQA, and NExT-QA origins. On the remaining disagreement cases, lookahead improves realized utility by 3.3, 3.1, and 4.6 percentage points. The Spearman correlation between candidate score and realized utility changes from 0.34 to 0.48 on GQA, 0.37 to 0.51 on SlideVQA, and 0.31 to 0.45 on NExT-QA.

C.3 Direct-Scoring Controls

The main direct scorer learns its own readout and scoring head from the same frozen backbone inputs and offline trajectories. We compare it with a variant that is forced to reuse the readout learned by the transition model. The independent version is slightly stronger and is therefore used as the main baseline.

Allowing the direct baseline to learn its own representation improves both downstream scores and same-origin action quality slightly, but does not remove the gap to lookahead. The main comparison therefore does not depend on forcing direct scoring to operate in the transition model's readout space.

Table A8: Same-origin realized utility with Qwen3-VL-8B-Instruct. Confidence intervals are computed from paired, grouped bootstrap resampling.

Env.	N	Direct	Lookahead	Δ	95% CI
GQA	600	.655	**.669**	+.014	$[-.001, .029]$
SlideVQA	600	.765	**.777**	+.012	$[-.002, .026]$
NExT-QA	600	.831	**.846**	+.015	$[.003, .027]$
Pooled	**1800**	.752	**.765**	**+.0138**	$[\mathbf{.005}, \mathbf{.022}]$

Table A9: Best-action agreement and empirical regret in the same-origin evaluation.

	Best-action match (%)		Regret ↓	
Environment	Direct	Lookahead	Direct	Lookahead
GQA	53.8	**62.1**	.029	**.015**
SlideVQA	55.1	**63.6**	.028	**.016**
NExT-QA	58.4	**67.0**	.028	**.013**
Pooled	55.9	**64.3**	.0283	**.0146**

D Representation and Planning Ablations

D.1 Readout Design

We vary pooling, latent dimension, backbone layer, and projector capacity while keeping the Qwen3-VL-8B backbone fixed. Predictive fidelity is averaged across the three environments. The downstream column reports the mean of GQA accuracy, SlideVQA joint F1, and NExT-QA accuracy.

Increasing readout capacity can improve latent prediction without improving action selection. The two-layer projector obtains the highest one-step cosine similarity in this comparison, while the simpler $d = 128$ mean-pooled linear readout gives the highest downstream mean. We use the latter throughout the main experiments.

D.2 Local Error Parameters

The local error term depends on neighborhood size k and penalty weight λ. The response is broad around the default values rather than sharply tuned to a single setting.

A single neighbor produces a noisier local estimate, whereas large neighborhoods smooth across less similar states. Removing the error penalty reduces performance on all three tasks. Large values of λ become overly conservative, while the range $.05$–$.20$ gives similar results.

D.3 Training-Data Scaling

The latent models are trained from offline trajectories collected by the frozen reactive policy. To measure how strongly the method depends on the amount of offline interaction data, we train the Qwen3-VL-8B latent module using progressively larger subsets of the available trajectories. The subset is applied jointly across the three environments, while the architecture, optimizer, and evaluation protocol are unchanged.

Predictive fidelity increases steadily as more trajectories are added, from $.822$ with one quarter of the data to $.868$ with the full training set. Downstream behavior follows the same broad trend, although it is not perfectly monotonic. SlideVQA, for example, reaches 76.12 with 50% of the trajectories and 76.05 with 75%, despite the corresponding increase in predictive fidelity. This is consistent with the main-text observation that better state prediction does not map to a fixed downstream improvement.

The scaling result also constrains how the learned transition model should be interpreted. Useful prediction is already present with a fraction of the available offline data, while additional trajectories improve the coverage of the state–action space and raise both fidelity and final performance on average. The full-data configuration is used in the main experiments, but the effect does not appear only after exposing the latent model to the complete trajectory collection.

Table A10: Direct-scoring controls with Qwen3-VL-8B-Instruct. The independent direct scorer is the version used in the main paper.

Method	GQA	SlideVQA	NExT-QA	Utility	Match (%)	Regret ↓
Direct, shared readout	63.88	75.68	83.71	.750	55.0	.0298
Direct, independent readout	64.10	75.90	83.85	.752	55.9	.0283
Latent Lookahead	**64.90**	**76.70**	**84.30**	**.765**	**64.3**	**.0146**

Table A11: Readout ablations with Qwen3-VL-8B-Instruct. The shaded row is the default configuration.

Readout	1-step cosine	Mean downstream
Last token, $d = 128$	.849	74.63
Mean, $d = 64$	.861	75.02
Mean, $d = 128$	.868	**75.30**
Mean, $d = 256$	.870	75.24
Attention pool, $d = 128$	.872	75.18
2-layer projector, $d = 128$	**.875**	75.16
Mean, layer -2	.861	75.01
Mean, layer -4	.847	74.52

D.4 Joint and Environment-Specific Training

The main configuration trains one latent module per backbone using trajectories from all three environments. We compare this setting with separate environment-specific transition and value modules. Apart from the partition of the training data, the model capacity and optimization settings are unchanged.

Environment-specific training gives slightly higher one-step fidelity on GQA and SlideVQA, improving cosine similarity from .889 to .892 and from .871 to .875, respectively. The ordering reverses on NExT-QA, where joint training reaches .844 compared with .842 for separate modules. These small predictive differences do not produce a consistent downstream advantage. Separate training is 0.06 points higher on SlideVQA, whereas joint training is 0.08 points higher on GQA and 0.12 points higher on NExT-QA.

The comparison is useful because joint training could otherwise act as an additional source of cross-environment supervision. The results show that this is not required for the predictive effect: separate modules recover nearly the same structure and even obtain slightly higher one-step fidelity in two environments. We retain joint training because it achieves comparable downstream performance while requiring only one latent module per backbone.

D.5 Residual-Cache Size

The local error estimate uses a cache of transition residuals collected during training. To separate this retrieval component from the learned dynamics, we subsample the cache after training while keeping the readout, transition model, and value head fixed.

Reducing the cache mainly affects the quality of the local reliability estimate. The correlation with realized transition error decreases from .50 with the full cache to .44 at 50% and .38 at 25%. Downstream degradation is more gradual. Retaining one quarter of the cache reduces GQA, SlideVQA, and NExT-QA by 0.31, 0.30, and 0.22 points relative to the full-cache setting. With half of the cache, the corresponding gaps are only 0.14, 0.15, and 0.09 points.

Because no latent-model parameters are retrained, these changes isolate the residual lookup used by $\epsilon(z, a)$. A denser cache improves the ranking of unreliable transitions, but the action-selection mechanism does not fail abruptly when local support is reduced. This suggests that the cache acts as a reliability guardrail rather than as an additional predictive model. Approximate retrieval or cache compression could therefore reduce storage cost in larger deployments without modifying the transition model itself.

Table A12: Sensitivity to the local-error neighborhood and penalty weight. Each block varies one parameter while fixing the other at its default.

Sweep	Value	GQA	SlideVQA	NExT-QA	Mean
k ($\lambda = .1$)	1	64.58	76.34	84.08	75.00
	5	64.82	76.60	84.21	75.21
	10	**64.90**	**76.70**	**84.30**	**75.30**
	20	64.87	76.66	84.19	75.24
	40	64.66	76.41	84.11	75.06
λ ($k = 10$)	0	64.49	76.25	83.98	74.91
	.05	64.78	76.55	84.17	75.17
	.10	**64.90**	**76.70**	**84.30**	**75.30**
	.20	64.84	76.61	84.14	75.20
	.40	64.30	75.98	83.77	74.68

Table A13: Effect of offline trajectory volume with Qwen3-VL-8B-Instruct. The second column reports one-step predictive fidelity averaged over the three environments.

Data	1-step	GQA	SlideVQA	NExT-QA
25%	.822	63.84	75.65	83.65
50%	.842	64.28	76.12	83.91
75%	.856	64.61	76.05	84.18
100%	**.868**	**64.90**	**76.70**	**84.30**

Table A14: Joint versus environment-specific latent-model training with Qwen3-VL-8B-Instruct.

Training	1-step cosine			Downstream		
	GQA	Slide	NExT	GQA	Slide	NExT
Separate	**.892**	**.875**	.842	64.82	**76.76**	84.18
Joint	.889	.871	**.844**	**64.90**	76.70	**84.30**

Table A15: Sensitivity to residual-cache size with Qwen3-VL-8B-Instruct. ρ is the pooled Spearman association between estimated and realized transition error.

Cache	ρ	GQA	SlideVQA	NExT-QA
25%	.38	64.59	76.40	84.08
50%	.44	64.76	76.55	84.21
100%	**.50**	**64.90**	**76.70**	**84.30**

E Additional Downstream Results

The main experiments report aggregate performance for each environment. Here we examine how the gains are distributed across question types and difficulty levels, and compare Latent Lookahead with task-specific evidence-routing methods. These analyses characterize where the downstream effect occurs; they are separate from the predictive diagnostics used to support the representation claim.

E.1 Task and Difficulty Breakdown

The aggregate gains conceal substantial variation across question types. On GQA, counting improves by 4.1 points, relation by 3.4, and spatial reasoning by 2.9, whereas existence and attribute questions gain 1.3 and 1.2 points. The larger changes occur on categories where locating or combining additional evidence can plausibly alter the answer, while categories often resolved from a single local observation leave less room for action selection to help.

SlideVQA exhibits a related pattern across its evaluation metrics. Token-level EM and F1 increase by 2.1 and 2.3 points, while joint EM and joint F1 both increase by 3.1 points. The larger change on the joint metrics is consistent with a benefit on examples where the final answer depends on recovering the complete set of required evidence. NExT-QA shows a smaller spread: causal questions improve by 1.9 points, temporal questions by 1.6, and descriptive questions by 1.1.

Table A16: Detailed downstream results with Qwen3-VL-8B-Instruct. Values are percentages.

Dataset	Metric / category	Reactive	Lookahead	Δ
GQA	Overall accuracy	62.7	**64.9**	+2.2
	Existence	83.1	**84.4**	+1.3
	Relation	60.3	**63.7**	+3.4
	Spatial	59.2	**62.1**	+2.9
	Counting	55.1	**59.2**	+4.1
	Attribute	64.3	**65.5**	+1.2
SlideVQA	EM	68.7	**70.8**	+2.1
	F1	79.9	**82.2**	+2.3
	Joint EM	65.4	**68.5**	+3.1
	Joint F1	73.6	**76.7**	+3.1
NExT-QA	Overall	82.7	**84.3**	+1.6
	Causal	82.5	**84.4**	+1.9
	Temporal	80.3	**81.9**	+1.6
	Descriptive	86.2	**87.3**	+1.1

Table A17: Difficulty and reasoning-type breakdown for GQA and SlideVQA.

Dataset	Split / type	Reactive	Lookahead	Δ
GQA	Easy (depth ≤ 2)	72.1	**73.1**	+1.0
	Medium (depth $= 3$)	61.2	**63.6**	+2.4
	Hard (depth ≥ 4)	50.1	**54.5**	+4.4
SlideVQA	Single-hop factual	76.1	**77.8**	+1.7
	Multi-hop / cross	68.2	**71.9**	+3.7
	Numerical reasoning	65.3	**70.1**	+4.8
	Layout / structure	72.2	**74.6**	+2.4
	Cross-slide compare	69.1	**72.7**	+3.6

These categories are not controlled experimental factors, so we do not attribute the gains to any single reasoning type. They instead indicate that the average improvement is distributed unevenly and tends to be larger on categories where the currently exposed observation is less likely to contain all required evidence.

The GQA depth split makes the relation with task difficulty more visible. Accuracy improves by 1.0 point on questions with program depth at most two, by 2.4 points at depth three, and by 4.4 points for depth four or greater. The reactive policy is also weakest on the hard split, so the result does not establish that program depth itself causes a larger lookahead gain. It does show that the aggregate improvement is not driven mainly by the easiest questions.

SlideVQA is less monotonic. Single-hop factual questions improve by 1.7 points, while multi-hop/cross examples and cross-slide comparisons gain 3.7 and 3.6 points. Numerical reasoning has the largest gain at 4.8 points, whereas layout/structure questions improve by 2.4. These categories vary in evidence distribution, answer structure, and reasoning demand, so the breakdown is better interpreted as a map of where additional visual access is useful than as a single-axis difficulty measure.

Latent Lookahead does not receive the category, hop count, or program depth as an explicit input to the planner. The concentration of larger gains in several harder and cross-evidence subsets therefore emerges from the visual-access decisions rather than from a category-specific policy component.

E.2 Task-Specific Evidence Routing

We also compare against evidence-routing methods designed for individual modalities. These baselines contain an explicit mechanism for deciding where to look, and therefore provide a stronger reference than the generic reactive policy.

Under matched interaction budgets, Latent Lookahead is 0.7 points above ViCrop on GQA-PatchEnv, 1.0 point above VDocRetriever on SlideVQA-Env, and 2.4 points above TOGA on NExT-

Table A18: Task-specific evidence-routing comparisons using Qwen3-VL-8B-Instruct.

Environment	Routing baseline	Baseline	Lookahead
GQA-PatchEnv	ViCrop	64.2	**64.9**
SlideVQA-Env	VDocRetriever	75.7	**76.7**
NExT-QA-Env	TOGA	81.9	**84.3**

QA-Env. The margin varies substantially across environments, and the table should not be read as a general ranking of these routing methods because their architectures and modality-specific inductive biases differ.

The comparison instead addresses whether the downstream gain can be explained solely by introducing an evidence-selection mechanism. Latent Lookahead remains competitive without using a separate routing architecture for image regions, presentation slides, and temporal video segments. The same transition-and-value formulation is used in all three environments.

Specialized routing and latent prediction are also not mutually exclusive. A routing system could propose a restricted set of candidate observations and a latent model could evaluate their predicted consequences. We do not combine them here because the resulting system would introduce task-specific components beyond the representation question studied in the paper.

F Efficiency and Profiling

We report both model-independent compute accounting and measured wall-clock behavior. Relative FLOPs describe how additional inference compute is allocated, while latency and memory capture implementation effects that FLOP counts do not represent. All measurements use the same complete-query accounting scope.

F.1 Compute Accounting

Relative FLOPs are normalized to a complete single-trajectory reactive query, including interaction-time VLM computation and final answer decoding. Latent transition and value computation is added for Latent Lookahead. Offline branching used by the same-origin diagnostic is not included because it is an evaluation procedure rather than deployed inference.

The compute sweep separates two ways of spending additional inference budget. Moving from the reactive policy to one-step latent lookahead raises relative FLOPs from $1.00\times$ to $1.15\times$ and gives gains of 0.30, 1.70, and 0.80 points on GQA, SlideVQA, and NExT-QA. At $H = 2$, the cost rises to $1.30\times$ and the gains increase to 2.20, 3.10, and 1.60 points. Extending the rollout to $H = 3$ requires $1.60\times$ compute, yet scores fall by 0.55, 0.52, and 0.28 points relative to $H = 2$.

This turnover agrees with the predictive analysis: longer latent rollouts continue to consume compute after predictive fidelity has started to deteriorate. The $H = 2$ setting should therefore be understood as the best operating point in the evaluated range, not as a general two-step optimum.

Multi-sample reactive inference allocates computation differently. Three complete trajectories require $3.0\times$ the reference compute and improve the task metrics by 0.60, 0.80, and 0.60 points. Five trajectories require $5.0\times$ compute and obtain gains of 1.10, 1.40, and 1.10 points. In the present experiments, $H = 2$ Latent Lookahead exceeds the five-trajectory reference on all three environments while using substantially less total query compute.

This comparison is not intended as a universal efficiency ranking. Repeated sampling and latent lookahead scale differently with backbone size, decoding length, and candidate-set size. It establishes a narrower result: under the frozen backbones and visual-access settings considered here, allocating a moderate amount of inference compute to predicted latent futures is more effective than allocating a much larger budget to independent repetitions of the same reactive policy.

F.2 Latency and Memory

FLOPs do not map directly to wall-clock time because the latent modules are much smaller than the VLM backbone and exhibit different hardware utilization. We therefore measure complete-query

Table A19: Compute–performance comparison with Qwen3-VL-8B-Instruct.

Method	FLOPs	GQA	Slide	NExT
Reactive	1.00×	62.70	73.60	82.70
Lookahead $H = 1$	1.15×	63.00	75.30	83.50
Lookahead $H = 2$	**1.30×**	**64.90**	**76.70**	**84.30**
Lookahead $H = 3$	1.60×	64.35	76.18	84.02
Reactive, 3 trajectories	3.00×	63.30	74.40	83.30
Reactive, 5 trajectories	5.00×	63.80	75.00	83.80

Table A20: Complete-query profiling on a single NVIDIA A100.

Backbone	Method	FLOPs	P50 ms	Memory GB
Qwen3-VL-8B	Reactive	1.00×	284	18.8
	Lookahead	**1.30×**	**346**	**19.1**
Phi-4-mm	Reactive	1.00×	255	15.6
	Lookahead	**1.29×**	**302**	**15.9**
Gemma-3n-E4B	Reactive	1.00×	223	12.2
	Lookahead	**1.32×**	**270**	**12.5**

latency separately on a single NVIDIA A100 with batch size one. Twenty complete queries are used for warm-up and the following 200 are timed. CUDA execution is synchronized immediately before and after each timing interval.

The wall-clock overhead is smaller than the nominal FLOP increase. For Qwen3-VL-8B, median latency rises from 284 to 346 ms, an increase of approximately 21.8%. Phi-4-mm increases from 255 to 302 ms (18.4%), and Gemma-3n-E4B from 223 to 270 ms (21.1%). By comparison, relative FLOPs increase by roughly 29–32%.

This difference is consistent with the structure of the added computation. Once the real VLM state has been encoded, imagined transitions are processed by a small transition MLP and value head rather than by additional backbone calls. Their arithmetic cost therefore does not reproduce the memory movement and decoding overhead of another full VLM trajectory. The precise relationship between FLOPs and latency remains implementation- and hardware-dependent, which is why both are reported.

Peak allocated memory increases by 0.3 GB for each backbone: 18.8 → 19.1 GB for Qwen3-VL-8B, 15.6 → 15.9 GB for Phi-4-mm, and 12.2 → 12.5 GB for Gemma-3n-E4B. The frozen backbone remains the dominant memory cost. The residual cache is separate from model parameters and, as shown in Table A15, can be subsampled when retrieval storage is more constrained.

G Additional Reproducibility Details

Latent-model training uses three fixed training seeds. Downstream evaluation uses three decoding seeds, with identical seed assignments across compared methods. Unless stated otherwise, ablation results use the same three decoding seeds as the corresponding main experiment. Predictive comparisons use identical held-out transition inputs and successor targets after the readout has been fixed.

For same-origin analysis, candidate branches from the same decision origin use three matched continuation seeds. Confidence intervals are computed from paired bootstrap differences. Resampling is grouped by the underlying image, slide deck, or video so that multiple decision origins from one visual instance are not treated as independent examples.

The hyperparameter studies use $d \in \{64, 128, 256\}$, $H \in \{1, 2, 3\}$, $k \in \{1, 5, 10, 20, 40\}$, and $\lambda \in \{0, .05, .10, .20, .40\}$ where applicable. Hyperparameter selection is performed on held-out training/validation data before evaluation. No evaluation instance is used for optimization or model selection.

The released implementation will include the environment wrappers, trajectory-collection scripts, latent-model configurations, residual cache construction, same-origin evaluation indices, and profil-

ing commands. Backbone checkpoints remain frozen throughout trajectory collection, latent-model training, and evaluation.