GAMEWISE: LEARNING WORLD-INDEXED SITUATED EXPERIENCE FOR ADAPTIVE DECISION-MAKING IN 3D GAMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual agents in unfamiliar 3D environments accumulate long interaction streams that are brittle to retrieve or replay as monolithic trajectories. We introduce GAMEWISE, a world-indexed situated experience reuse framework in which visually re-identifiable places partition long-horizon interaction into locally retrievable and verifiable segments. To determine where skill experience remains behaviorally reusable, *Skill-Conditioned Cross-Place Outcome Compatibility Sharing (SCOCS)* compares skill-conditioned empirical outcome distributions using optimal transport and recursively evaluates finite-depth successor compatibility under matched observable conditions. Screened successful segments adapt a visual–action controller to take over high-frequency low-level actions, while the VLM uses retrieved spatial and outcome evidence to issue persistent goals and replan on events. Under a common protocol in ViZDoom, GAMEWISE achieves the strongest system-level results. On held-out Xonotic and Sauerbraten, GAMEWISE acquires and reuses situated experience online, achieving the best task performance without target-engine policy training.

1 INTRODUCTION

Pretrained visual–action policies acquire broad behavioral priors from demonstrations (Baker et al., 2022; Reed et al., 2022; Yue et al., 2026; Magne et al., 2026), while memory-augmented agents use interaction history to support later decisions (Pritzel et al., 2017; Wang et al., 2024; Zhao et al., 2024; Tan et al., 2024). In an unfamiliar 3D world, however, interaction accumulates as long first-person streams. Matching or replaying an entire stream is brittle: a local viewpoint change or execution deviation can invalidate the whole trajectory, while sparse terminal outcomes provide little indication of which stage has been reached. We ask how an RGB-only agent can divide such experience into reusable units, locate its current stage, and determine which units remain applicable, without engine pose, a metric map, or privileged state.

Our central design hypothesis is that visually re-identifiable places provide natural retrieval and verification boundaries for long-horizon experience, while local state and skill-conditioned outcomes determine the applicability of the resulting segments. Doorways, junctions, rooms, and cover boundaries delimit stages that can be retrieved and visually verified independently. Indexing experience by these places localizes retrieval and comparison to a bounded part of history; recognized places and transitions also provide intermediate progress feedback about the current stage and expected successor. Place identity is not, however, a complete control state. Entering the same health-pack room at full versus low health, or the same doorway with different weapons, can change which experience applies. Conversely, the same skill can have compatible outcomes at distinct places that must remain separate for navigation. Thus spatial identity should segment and locate experience, whereas state- and skill-conditioned evidence should govern its reuse.

We introduce GAMEWISE (*World-Indexed Situated Experience*), an RGB-only system that realizes this decomposition through three complementary components. First, a pose-free *world index* partitions interaction into action–observation traces between re-identifiable places, supporting localized retrieval, progress tracking, and endpoint verification. Second, *SCOCS* reasons over accumulated spatial experience to determine, for each skill, which place-conditioned experiences exhibit differ-

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.

ent consequences and which remain behaviorally compatible enough to be reused across regions under the agent's current observable state. Third, successful situated segments that satisfy outcome, spatial-consistency, and target-validity checks are retained as goal-conditioned demonstrations for adapting the *VLA*, which takes over high-frequency low-level control from the VLM. The three components therefore address complementary aspects of situated experience: the world index determines where experience belongs, SCOCS determines where it remains behaviorally reusable, and VLA adaptation consolidates reusable experience into efficient execution.

Deciding whether skill experience remains compatible across places from sparse RGB-only interaction, rather than from a well-sampled Markov state space, requires a different notion of behavioral compatibility. SCOCS instantiates this notion by adapting the behavioral-similarity principle of bisimulation (Ferns et al., 2012; Gelada et al., 2019; Zhang et al., 2020; Castro et al., 2021): it replaces reward-based state equivalence with skill-conditioned empirical state changes, compares outcome distributions using optimal transport, and recursively evaluates finite-depth successor compatibility. This enables cross-place reuse of skill experience.

We evaluate GAMEWISE on navigation and combat in ViZDoom, with cross-engine tests in Xonotic and Sauerbraten. GAMEWISE-Hybrid achieves the best task performance among the evaluated methods across all three games: ViZDoom, Xonotic, and Sauerbraten (Tables 1 and 2). Controlled comparisons examine spatial indexing, skill-experience organization, and module interactions (Tables 3–5; Appendix Table C.2). On held-out engines, adaptation improves the VLA over its pre-adaptation checkpoint on every reported task metric, while the hybrid with memory built online in each target engine performs best among the evaluated variants (Table 2).

Our contributions are:

- **World-indexed segmentation and reuse of long-horizon experience.** Visually re-identifiable places partition RGB interaction streams into locally retrievable and verifiable segments, providing intermediate progress feedback while separating spatial identity from state- and skill-conditioned applicability.
- **Skill-conditioned cross-place outcome compatibility sharing.** SCOCS compares empirical outcomes and finite-depth successors under matched observable conditions. Missing-evidence bounds and skill-specific complete-link groups enable cross-place consultation without merging spatial nodes or asserting global state equivalence.
- **Task-dependent decomposition of situated reuse and adaptation.** Controlled comparisons show that spatial indexing and outcome sharing contribute to different task types—route retrieval versus resource- and state-dependent combat—and that selectivity rather than sharing coverage drives the gain; cross-engine experiments separate behavior consolidated into the VLA from memory constructed online in each target world.

2 RELATED WORK

Game policies and language-guided agents. VPT and STEVE-1 learn reusable behavior from video in Minecraft (Baker et al., 2022; Lifshitz et al., 2023), while OpenP2P and NitroGen scale video–action learning across many games (Yue et al., 2026; Magne et al., 2026). SIMA studies language-conditioned keyboard–mouse control (Raad et al., 2024), and Voyager combines language planning with a programmatic skill library (Wang et al., 2023). SayCan and LM-Nav ground language in skills and landmarks (Ahn et al., 2022; Shah et al., 2023); RT-2 learns tokenized visual control (Brohan et al., 2023). Our focus is how an RGB-only agent organizes and reuses its own spatially situated experience.

Experience and spatial memory. JARVIS-1, ExpeL, RAP, and AutoManual retrieve episodes or induced knowledge to improve later decisions (Wang et al., 2024; Zhao et al., 2024; Kagaya et al., 2024; Chen et al., 2024). Metric memories organize observations spatially (Parisotto & Salakhutdinov, 2017; Gupta et al., 2017; Chaplot et al., 2020a; Huang et al., 2023); topological memories retain navigable connectivity (Savinov et al., 2018; Shah et al., 2021; Chaplot et al., 2020b). 3D-Mem, Embodied-RAG, and MemoNav further enrich retrievable scene memory (Yang et al., 2025; Xie et al., 2024; Li et al., 2024). These works provide complementary mechanisms for retrieving useful history and organizing visited scenes. Our focus is jointly representing where an interaction

occurred and whether a skill can be reused under the current agent state, and sharing its empirical outcomes without merging places.

Behavioral distances, optimal transport, and policy transfer. Bisimulation metrics compare states by reward and transition similarity through optimal transport (Ferns et al., 2012; 2011), with sampling-based variants that avoid explicit couplings (Castro et al., 2021) and learned latent instantiations that preserve decision-relevant structure (Zhang et al., 2020; Gelada et al., 2019). SCOCS adopts the recursive comparison of successors from this line of work under three restrictions suited to an agent's own early experience: it uses observed outcomes in place of reward, compares one skill at a time instead of maximizing over all actions, and terminates at finite depth on an observation projection, with optimal transport aligning experience sets of different sizes. Concrete place identities and original records remain available for navigation, retrieval, and visually grounded execution. DAgger and distillation transfer supervision to learned policies (Ross et al., 2011; Hinton et al., 2015); our VLA post-training is a downstream use of collected experience rather than the sharing criterion itself.

3 Problem Formulation

The agent receives RGB images o_t, a task instruction x, and its prior keyboard–mouse actions u_t, but no engine pose or privileged state. Skills execute over multiple low-level steps (Sutton et al., 1999) and are predefined; skill discovery is outside our scope. We study how to organize and reuse executed skill experience across places under fixed interaction and memory budgets.

Let s denote an observation-derived state that associates the agent's current place with observable conditions relevant to experience reuse. It is a partial description, not the full environment state (Kaelbling et al., 1998). Executing skill a produces an observed outcome ω and successor state s', yielding an experience record

$$e = (s, a, \omega, s'). \tag{1}$$

The concrete state and outcome fields depend on the environment; our experimental instantiation is specified in Appendix A.2.

4 Method

4.1 World-Indexed Situated Experience Reuse

Places determine where long experience is segmented and located; state- and skill-conditioned outcomes determine when a segment is reusable. GameWISE therefore retains two independent structures. A pose-free spatial graph partitions the interaction stream into behavioral transitions between concrete, re-identifiable places, supporting local retrieval and stage-level verification. SCOCS then enables cross-place experience reuse by sharing skill-specific records with compatible outcomes. The system first relocalizes and retrieves experience by state and skill; the VLM combines this evidence with current RGB to select and ground a local goal, which the VLA persistently executes. Actual outcomes update memory, and locally valid segments are available for downstream adaptation (Figure 1). Section 4.2 addresses spatial indexing, Section 4.3 defines sharing, and Section 4.4 connects goal-conditioned control.

4.2 Spatial Indexing and State Conditions

Rather than storing an episode as one monolithic video, the world index uses recognized places as boundaries and records the action–observation stream between successive place scans:

$$\begin{aligned} &\mathcal{M} = (\mathcal{V}, \mathcal{E}), && e_{ij} = (v_i, v_j, \tau_{ij}, q_{ij}), \\ &\tau_{ij} = (o_0, u_0, o_1, \ldots, u_{L-1}, o_L). \end{aligned} \tag{2}$$

Place v_i is represented by multi-view RGB, τ_{ij} is the resulting local segment, and q_{ij} is its quality status. This representation bounds retrieval to nearby segments and turns re-identified places and verified transitions into intermediate progress landmarks. An edge establishes that a behavioral transition occurred, not that it is always traversable.

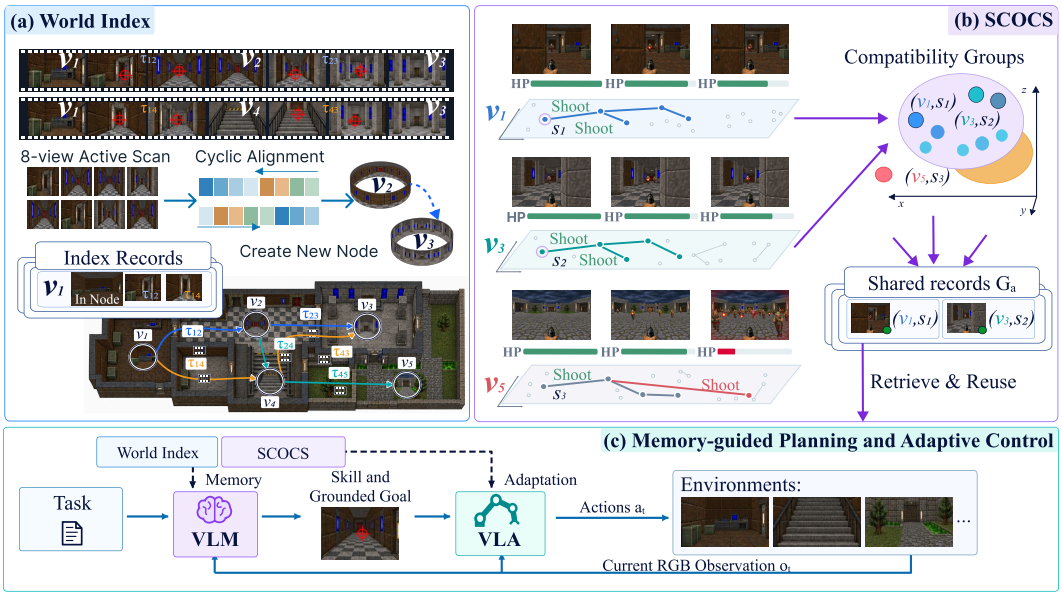

Figure 1: Overview of GAMEWISE. Visual exploration builds pose-free spatial memory and spatially anchored skill experience. SCOCS enables cross-place experience reuse by sharing skill-specific records with compatible outcomes; screened interaction trajectories support VLA adaptation. At deployment, the VLM retrieves experience for low-frequency decisions, and the VLA executes persistent instructions through high-frequency visual control.

Unfamiliar visual styles and interaction rules can cause the VLM to misidentify exits, obstacles, or resources. Its proposed landmarks and frontiers therefore only guide exploration. On revisits, multi-view matching and geometric verification (Rublee et al., 2011; Fischler & Bolles, 1981) recover place identity; insufficient evidence or ambiguous candidates return the unknown place $\perp$ rather than forcing an association.

Given localization $\hat{v}_t$, candidate executions must satisfy the quality requirement:

$$\mathcal{E}_t = \{e \in \mathcal{E} : \mathrm{src}(e) = \hat{v}_t,\ q_e \in \mathcal{Q}_{\mathrm{allowed}}\}. \tag{3}$$

Reuse additionally verifies that current RGB satisfies the historical starting conditions. Execution is checked and corrected step by step; a mismatch triggers exit and relocalization. **Finishing an action sequence does not mean completing a spatial transition**: endpoint visual verification and target-place re-identification jointly confirm handoff. Active scanning, matching, correction, and handoff rules appear in Appendix A.1.1. Reuse also depends on the observable conditions represented by s. Since this state description is partial, the VLM still uses current RGB to assess historical evidence. Spatial indexing locates candidate experience; the next section determines whether a skill's records can be shared across places.

4.3 SKILL-CONDITIONED CROSS-PLACE OUTCOME COMPATIBILITY SHARING (SCOCS)

Spatial indexing preserves local differences; outcome compatibility determines when experience can be shared. The same skill under the same observable conditions can have different consequences at different places: firing from a doorway may expose the agent differently from firing in an open square. We therefore accumulate experience separately by place, condition, and skill, rather than mixing records across places in advance. Conversely, distinct places may yield compatible outcomes and subsequent behavior for a particular skill. SCOCS compares their separately accumulated evidence and pools that skill's records only when they pass the compatibility criterion below. Spatial identity thus distinguishes the sources of experience, while behavioral comparison determines which sources can supplement one another.

Write the observation-derived state as $s = (p, z)$, where p identifies a place in the spatial graph and z contains the non-spatial conditions used for sharing. Each exact state–skill key $((p, z), a)$ indexes

its own record set $\mathcal{D}((p, z), a)$. For skill a, candidate states $s_i = (p_i, z)$ and $s_j = (p_j, z)$ must have known places, matching conditions z, and records of a on both sides; p_i and p_j need not be equal. In our experiments, z combines HUD information read by manually calibrated OCR with shared auxiliary visual features collected before each decision; health, armor, and ammunition are discretized into ten levels (Appendix A.2.1).

Let $\widehat{P}_{s,a}$ be the equal-weight empirical distribution of outcome–successor pairs (ω, s') in $\mathcal{D}(s, a)$. We compare $\widehat{P}_{(p_i,z),a}$ and $\widehat{P}_{(p_j,z),a}$ while keeping their source record sets separate. Let $\delta_\omega(\omega_i, \omega_j) \in [0, 1]$ measure immediate outcome discrepancy. Then

$$\begin{aligned} c_h(e_i, e_j) &= \max\{\delta_\omega(\omega_i, \omega_j), \gamma d_{h-1}(s'_i, s'_j)\}, \\ D^a_h(s_i, s_j) &= \min_{\pi \in \Pi(\widehat{P}_{s_i,a}, \widehat{P}_{s_j,a})} \sum_{u,v} \pi_{uv}\, c_h(e_u, e_v). \end{aligned} \tag{4}$$

Here Π is the set of couplings between the two empirical distributions. Optimal transport (Gabriel & Marco, 2019) compares distributions with different sample counts rather than only their means; recursion checks whether similar immediate outcomes lead to states with compatible subsequent behavior under the same skills. Depth h counts skill segments, with successor distance

$$d_h(s_i, s_j) = \max\left\{ d_0(s_i, s_j), \max_{a \in A_{\text{obs}}(s_i) \cup A_{\text{obs}}(s_j)} D^a_h(s_i, s_j) \right\}. \tag{5}$$

The base distance d_0 compares non-spatial state attributes: different place IDs alone incur no penalty. Place identity remains in each state key and determines which records supply its empirical distributions; it is not erased before comparison. The recursion compares observed skill outcomes and successors, rather than matching spatial topology or route geometry. Source states need only share observations of the current skill, whereas successor comparisons use the union of observed skills. The environment-specific definitions of d_0 and δ_ω appear in Appendix A.2.

Among candidate states satisfying the matching conditions above, each skill independently forms complete-link groups G_a such that

$$\max_{s_i, s_j \in G_a} \overline{D}^a_h(s_i, s_j) \le \varepsilon. \tag{6}$$

Here $\overline{D}^a_h$ is the conservative distance upper bound defined in Appendix A.2.2. Every pair in a group must pass this test, preventing chaining from "A resembles B and B resembles C." Only then are the skill's records pooled as $\mathcal{D}(G_a, a) = \biguplus_{s \in G_a} \mathcal{D}(s, a)$ for retrieval. A state can share Pickup experience with a state at place B and Combat experience with one at place C. These are skill-specific sharing relations, not a unified state equivalence class: original records retain their source and successor states, and spatial nodes and topology remain unchanged.

Queries start from the current place-indexed state $s_t = (p_t, z_t)$ and retrieve each skill's records from the group containing that state. Unknown places or missing exact state keys do not borrow across places. Shared summaries provide historical evidence from compatible sources and, together with spatial memory and current RGB observations, inform the VLM's decisions. Grouping parameters appear in Appendix A.2; updates, fallback order, and summary interfaces appear in Appendix A.2.3.

4.4 From Experience Reuse to Goal-Conditioned Execution

Spatial memory and SCOCS not only inform online decisions but also support the collection of goal-conditioned demonstrations, allowing useful behavior to transfer across environments through the controller. During interaction, the VLM uses spatial memory, SCOCS summaries, and current RGB to select skills and ground local targets. We record the resulting RGB–action segments with their target conditions, retaining successful segments that pass outcome verification, spatial-consistency, and target-validity checks. These demonstrations adapt a pretrained VLA through goal-conditioned behavior cloning (Pomerleau, 1991), teaching it to predict low-level actions from RGB and grounded targets. At deployment, the adapted VLA keeps fixed. In GameWISE-Hybrid, the VLM is invoked every 16 environment steps, compared with every 4 steps in GameWISE-VLM and the VLM baselines; the VLA executes the persistent target condition between planner calls. Skill completion, target invalidation, or relevant events trigger replanning. Data screening, target encoding, and training losses are detailed in Appendix A.3.

Table 1: ViZDoom system-level results. Bold marks the numerically best value per column (lowest for p95 latency). The efficiency column averages p95 step latency over the three scenarios.

	Deathmatch		My Way Home		Defend Center	Efficiency
Method	Reward ↑	Kills ↑	SR % ↑	SPL ↑	Reward ↑	p95 s ↓
VLM-based agents (shared planner, executors, budget)						
No memory	5.10	2.05	0	0.000	0.35	4.94
ReAct + Reflexion (Yao et al., 2022; Shinn et al., 2023)	2.40	1.15	5	0.013	0.95	10.11
SPTM (Savinov et al., 2018)	7.40	3.45	40	0.310	1.05	9.49
ExpeL (Zhao et al., 2024)	5.10	2.15	5	0.022	0.75	11.06
RAP (Kagaya et al., 2024)	6.30	2.70	0	0.000	0.55	10.67
JARVIS-1 (Wang et al., 2024)	7.10	3.10	0	0.000	3.20	9.50
VLA-based video policies						
OpenP2P (Yue et al., 2026)	7.66	3.25	10	0.047	9.65	0.26
NitroGen (Magne et al., 2026)	6.00	3.35	0	0.000	0.65	**0.19**
Ours						
GAMEWISE-VLM	13.24	5.75	65	0.577	12.25	5.52
GAMEWISE-VLA	7.33	5.20	35	0.280	6.45	0.32
GAMEWISE-Hybrid	**15.33**	**6.45**	**75**	**0.583**	**14.65**	1.30

Table 2: Cross-engine results on held-out Xonotic and Sauerbraten, reported as means. Bold marks the numerically best value per column (lowest for p95 latency).

	Xonotic			Sauerbraten			Efficiency
Method	Kills ↑	Coverage ↑	Travel Dist. ↑	Kills ↑	Coverage ↑	Travel Dist. ↑	p95 s ↓
SPTM	0.35	0.43	76.12	1.15	0.48	68.11	7.91
JARVIS-1	0.25	0.23	32.15	1.40	0.38	32.33	9.28
GAMEWISE-VLM	1.45	0.53	102.49	2.50	0.58	88.21	3.70
OpenP2P	0.00	0.12	24.25	0.11	0.07	21.20	**0.28**
GAMEWISE-VLA$_{\text{pre-adapt}}$	0.00	0.17	26.51	0.15	0.07	23.19	0.31
GAMEWISE-VLA	0.60	0.44	58.70	0.75	0.25	67.31	0.31
GAMEWISE-Hybrid	**2.15**	**0.76**	**123.68**	**3.55**	**0.65**	**102.43**	0.90

5 EXPERIMENTS

5.1 IMPLEMENTATION DETAILS

All VLM-based agents share one frozen, locally deployed Qwen3.5-9B planner at temperature 0. The VLM baselines and GAMEWISE-VLM use the same three rule-based executors (Explore, Pickup, Combat) and invoke the planner every 4 environment steps. Only GAMEWISE-Hybrid invokes its VLM every 16 environment steps, using the adapted VLA for low-level execution. All VLM requests block until a response or timeout, and the waiting time is charged to the requesting step. Our VLA adopts the Qwen3.5-0.8B architecture with action tokens and a decoder. Architecture and loss definitions are provided in Appendix A.3; training settings appear in Appendix B.1.

5.2 EVALUATION PROTOCOL

We evaluate ViZDoom (Kempka et al., 2016) navigation and combat in My Way Home, Deathmatch, and Defend the Center, with Xonotic and Sauerbraten used for cross-engine evaluation. Each cell is evaluated for 20 episodes with seeds 700–719, and we report per-cell means. Every environment begins with empty memory under a fixed-budget interaction–reuse protocol, and memory variants are compared on shared histories before their online trajectories diverge. Models receive RGB, with task text supplied to the VLM planners; standalone GAMEWISE-VLA acts from RGB without task-language, skill, or grounding inputs. Engine state is restricted to offline metrics under the standard

Table 3: Ablation over spatial indexing, SCOCS skill-experience sharing, and the persistent VLM–VLA execution interface. Metrics follow Table 1.

Enabled mechanism			Deathmatch		My Way Home		Defend Center
Spatial Indexing	SCOCS Sharing	VLM–VLA Interface	Reward ↑	Kills ↑	SR % ↑	SPL ↑	Reward ↑
			5.10	2.05	0	0.000	0.35
✓			6.67	3.65	60	0.529	7.20
	✓		7.32	4.25	20	0.139	7.30
✓	✓		13.24	5.75	65	0.577	12.25
		✓	5.00	2.00	15	0.145	8.35
✓		✓	5.67	3.65	65	**0.589**	6.65
	✓	✓	10.67	5.65	15	0.109	12.65
✓	✓	✓	**15.33**	**6.45**	**75**	0.583	**14.65**

Table 4: World-index mechanism ablation on the frozen memory under the GAMEWISE-VLM configuration. The planner, executors, and interaction budget are fixed.

Representation	Place R@1 ↑	False merge ↓	Dupl. node ↓	MWH SR % ↑	MWH SPL ↑
Bounded recent history	5.5%	–	–	10	0.041
Generic visual-keyframe topology	**60.7%**	5.3%	42.9%	25	0.179
VLM anchors, no visual reassociation	4.5%	81.8%	**6.5%**	15	0.093
GAMEWISE world index	58.5%	**3.7%**	18.3%	65	**0.577**

protocol; the Q-learning reference in Section 5.4.2 additionally uses Deathmatch reward labels to construct its Q-value context. Memory is never transferred between environments. For My Way Home, success rate measures how often the goal is reached, while SPL (Anderson et al., 2018) weights success by shortest-path efficiency, separating navigation reliability from route quality. For Deathmatch, cumulative scenario reward measures overall task performance and kills count defeated opponents, jointly assessing task return and combat effectiveness. For Defend the Center, reward summarizes kills and deaths under the scenario's fixed scoring rule to assess sustained defense, so a separate kill column is omitted. For Xonotic and Sauerbraten, kills, coverage (the spatial extent explored), and travel distance (cumulative movement over a fixed horizon) assess combat, exploration, and mobility under cross-engine shifts. Across games, p95 step latency is the 95th percentile of step execution time, measuring responsiveness including planner delays. Appendix B.2 provides the full task, metric, reuse, and fairness protocols.

5.3 COMPARISON WITH EXISTING METHODS

We compare representative memory mechanisms within a shared VLM harness, covering topological spatial memory represented by Semi-Parametric Topological Memory (SPTM) (Savinov et al., 2018) and major experience-memory families represented by ReAct + Reflexion, ExpeL, RAP, and JARVIS-1 (Yao et al., 2022; Shinn et al., 2023; Zhao et al., 2024; Kagaya et al., 2024; Wang et al., 2024), together with a memoryless control. These VLM memory variants and GAMEWISE-VLM share the planner, four-step decision interval, rule-based executors, action space, and interaction budget; GAMEWISE-Hybrid instead uses a sixteen-step VLM interval and VLA execution. We also compare OpenP2P and NitroGen VLA architectures (Yue et al., 2026; Magne et al., 2026) and report GAMEWISE-VLM, GAMEWISE-VLA, and GAMEWISE-Hybrid. Method-specific implementations are detailed in Appendix B.1.

5.3.1 MAIN RESULTS

ViZDoom comparison. Table 1 first shows that prompt-level memory is task-dependent and often insufficient. JARVIS-1 is the strongest baseline on Defend the Center, where its situation and plan memory matches a hold-and-patrol structure. On Deathmatch, by contrast, no prompt-level mechanism separates from the memoryless control under seed-paired testing: the largest nominal gain is $+2.30$ reward for SPTM, and the apparent deficit of ReAct + Reflexion is driven by a small

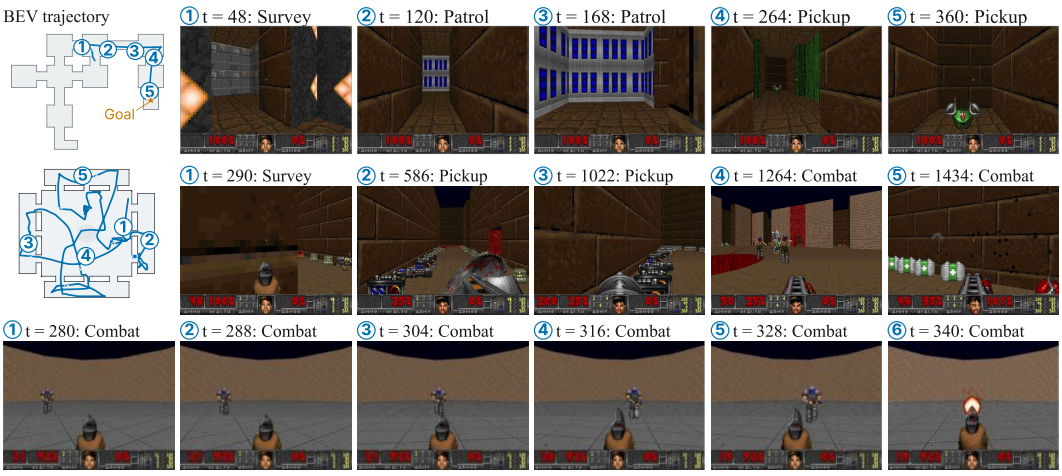

Figure 2: Representative GAMEWISE trajectories in ViZDoom. In My Way Home, the agent recalls a matched place, expands the spatial memory, and approaches the goal. In Deathmatch, the agent proactively switches between collecting items and attacking enemies based on its current state.

Table 5: Experience organization on ViZDoom Deathmatch under GAMEWISE-VLM. The experience library, spatial index, planner, executors, and retrieval token cap are shared. Support/query counts the historical records underlying the retrieved entries, not the number of entries; we report its mean with a 95% CI.

Method	Reward ↑ (95% CI)	Exp. tokens ↓ (95% CI)	Non-empty retrieval	Support/query ↑ (95% CI)
State-graph experience	8.50 ± 2.71	512.02 ± 15.08	**88.68%**	2.68 ± 0.07
Semantic clustering	10.35 ± 3.59	520.53 ± 23.76	79.00%	3.70 ± 0.28
Q-learning	9.55 ± 3.73	$\mathbf{203.69 \pm 6.51}$	67.05%	1.92 ± 0.13
SCOCS	$\mathbf{13.24 \pm 1.60}$	362.54 ± 56.20	87.35%	$\mathbf{8.46 \pm 1.32}$

number of episodes rather than a consistent shift. The memory baselines also show higher p95 step latency than the memoryless control; the aggregate metric does not isolate the costs of longer prompts, retrieval, or memory updates. GAMEWISE-VLM exceeds every VLM baseline on all task metrics, and GAMEWISE-Hybrid attains the highest Deathmatch reward and kills, My Way Home success rate and SPL, and Defend the Center reward. The higher success rate and SPL on My Way Home suggest that retrieving and visually verifying place-indexed route segments supports more reliable goal reaching and more efficient navigation. Figure 2 illustrates representative navigation and combat trajectories.

Cross-engine comparison. Table 2 evaluates held-out engines without target-engine policy training. Starting from empty memory, GAMEWISE-VLM acquires and reuses spatially anchored experience online rather than relying on a transferred map. Experience adaptation improves the VLA over its pre-adaptation checkpoint on every task metric in both engines, supporting the downstream utility of the screened interaction segments. Combining the adapted controller with online memory is strongest among the listed variants: GAMEWISE-Hybrid leads every task metric while reducing p95 latency from 3.70 s for the VLM scaffold to 0.90 s. Figure 3 shows representative behavior sequences in Sauerbraten and Xonotic.

5.4 ABLATIONS AND ANALYSIS

System-level ablation. Table 3 evaluates all combinations of spatial indexing, SCOCS skill-experience sharing, and the persistent execution interface. Disabling spatial indexing turns off spatial-memory use and navigation while retaining the place identities required to construct SCOCS state keys. Spatial indexing has its largest effect on navigation, raising My Way Home SPL from 0



Figure 3: Representative trajectories in the held-out Sauerbraten and Xonotic engines. The selected sequences illustrate resource acquisition, engagement, reorientation, interception, and recovery under changes in rendering and dynamics.

to 0.529; skill-experience sharing chiefly improves combat, raising Deathmatch reward from 5.10 to 7.32 when used alone. These task-dependent effects are consistent with the distinction between locating experience and assessing whether a particular skill's experience is reusable.

5.4.1 RQ1: How Does Spatial Indexing Support Experience Reuse?

Table 4 isolates the spatial representation under matched budgets. Generic keyframe topology attains the highest place R@1 but produces many duplicate nodes (42.9%), while VLM anchors without visual reassociation cause an 81.8% false-merge rate. Combining semantic anchors with visual reassociation reduces false merges to 3.7% and achieves the best SPL, despite a slightly lower R@1 than generic topology. The results show that place-retrieval accuracy alone does not capture the usefulness of a spatial index: duplicate identities fragment experience, whereas false associations can attach records to the wrong place. This comparison supports visually verified indexing for reuse.

5.4.2 RQ2: How Should Graph-Indexed Experience Be Organized for Reuse?

Table 5 compares four organizations of the same experience library. Direct state-graph retrieval preserves spatial specificity, but leaves experience for potentially shareable actions fragmented across local regions of the world index. Despite its 88.68% non-empty retrieval rate, it retrieves only 2.68 supporting records per query. Semantic clustering increases support to 3.70 by grouping semantically similar places, but place-level similarity does not distinguish agent states: experience collected with high health or a different weapon may be inappropriate on a later visit. More support therefore need not imply more applicable evidence. The Q-learning reference (Watkins & Dayan, 1992) supplies per-skill Q-values using additional Deathmatch reward labels. It produces the shortest context, but retrieves only 1.92 supporting records per query; compact value summaries alone do not resolve sparse evidence. SCOCS instead pools skill-specific support across places only under matched observable conditions and compatible outcomes, without merging spatial nodes or discarding record provenance. It retrieves 8.46 supporting records per query, 3.16× that of direct state-graph retrieval attaining the highest mean reward. These results support the benefit of preventing reusable action experience from being confined to separate local regions: the gain is in the supporting evidence available per query, rather than a higher non-empty retrieval rate. Further experiments on sharing-threshold sensitivity and SCOCS component ablations are reported in Appendix C.1.

6 Conclusion

We introduced GameWISE for world-indexed situated experience reuse: visual place identity determines where experience occurred, while state- and skill-conditioned outcomes determine where it can be reused. SCOCS uses optimal transport to compare immediate outcomes and finite-depth successors on jointly observed skills. Persistent instructions carry VLM skill decisions into fast visual control. The design addresses both fragmentation from strictly place-local storage and dynamic-state conflation from merging entire places.

AI USE STATEMENT

We used generative AI tools to assist with data visualization and figure creation, literature research, and drafting the initial manuscript. All manuscript content was subsequently revised and reviewed in full by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work studies RGB-only agents in virtual game environments. We do not identify specific ethical risks associated with the research described in this paper.

REPRODUCIBILITY STATEMENT

Details supporting reproducibility are provided in the appendix. Appendix A describes spatial indexing, SCOCS, and visual–action adaptation and execution. Appendix B provides implementation and training settings, evaluation protocols, and statistical testing procedures. Appendix C reports sharing-threshold sensitivity, SCOCS component ablations, and seed-paired comparisons with memory baselines.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Pablo Samuel Castro, Tyler Kastner, Prakash Panangaden, and Mark Rowland. Mico: Improved representations via sampling-based state similarity for markov decision processes. *Advances in Neural Information Processing Systems*, 34:30113–30126, 2021.

Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*, 2020a.

Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological slam for visual navigation. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 12872–12881. IEEE, 2020b.

Minghao Chen, Yihang Li, Yanting Yang, Shiyu Yu, Binbin Lin, and Xiaofei He. Automanual: Constructing instruction manuals by llm agents via interactive environmental learning. *Advances in Neural Information Processing Systems*, 37:589–631, 2024.

Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

Norman Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. *arXiv preprint arXiv:1207.4114*, 2012.

Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

Peyré Gabriel and Cuturi Marco. Computational optimal transport with applications to data sciences. *Foundations and trends® in machine learning*, 11(5-6):355–607, 2019.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International conference on machine learning*, pp. 2170–2179. PMLR, 2019.

Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2616–2625, 2017.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10608–10615. IEEE, 2023.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Tomoyuki Kagaya, Thong Jing Yuan, Yuxuan Lou, Jayashree Karlekar, Sugiri Pranata, Akira Kinose, Koki Oguri, Felix Wick, and Yang You. Rap: Retrieval-augmented planning with contextual memory for multimodal llm agents. *arXiv preprint arXiv:2402.03610*, 2024.

Michał Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaśkowski. Vizdoom: A doom-based ai research platform for visual reinforcement learning. In *2016 IEEE conference on computational intelligence and games (CIG)*, pp. 1–8. IEEE, 2016.

Hongxin Li, Zeyu Wang, Xu Yang, Yuran Yang, Shuqi Mei, and Zhaoxiang Zhang. Memonav: Working memory model for visual navigation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17913–17922. IEEE, 2024.

Shalev Lifshitz, Keiran Paster, Harris Chan, Jimmy Ba, and Sheila McIlraith. Steve-1: A generative model for text-to-behavior in minecraft. *Advances in Neural Information Processing Systems*, 36: 69900–69929, 2023.

Loïc Magne, Anas Awadalla, Guanzhi Wang, Yinzhen Xu, Joshua Belofsky, Fengyuan Hu, Joohwan Kim, Ludwig Schmidt, Georgia Gkioxari, Jan Kautz, et al. Nitrogen: An open foundation model for generalist gaming agents. *arXiv preprint arXiv:2601.02427*, 2026.

Emilio Parisotto and Ruslan Salakhutdinov. Neural map: Structured memory for deep reinforcement learning. *arXiv preprint arXiv:1702.08360*, 2017.

Dean A Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural computation*, 3(1):88–97, 1991.

Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adria Puigdomenech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In *International conference on machine learning*, pp. 2827–2836. PMLR, 2017.

Maria Abi Raad, Arun Ahuja, Catarina Barros, Frederic Besse, Andrew Bolt, Adrian Bolton, Bethanie Brownfield, Gavin Buttimore, Max Cant, Sarah Chakera, et al. Scaling instructable agents across many simulated worlds. *arXiv preprint arXiv:2404.10179*, 2024.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pp. 2564–2571. Ieee, 2011.

Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. *arXiv preprint arXiv:1803.00653*, 2018.

Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Ving: Learning open-world navigation with visual goals. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13215–13222. IEEE, 2021.

Dhruv Shah, Błażej Osiński, Sergey Levine, et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on robot learning*, pp. 492–504. pmlr, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

Weihao Tan, Wentao Zhang, Xinrun Xu, Haochong Xia, Ziluo Ding, Boyu Li, Bohan Zhou, Junpeng Yue, Jiechuan Jiang, Yewen Li, et al. Cradle: Empowering foundation agents towards general computer control. *arXiv preprint arXiv:2403.03186*, 2024.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023. *URL https://arxiv. org/abs/2305.16291*, 2(11), 2023.

Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jinbing Hou, Bowei Zhang, Haowei Lin, Zhaofeng He, Zilong Zheng, Yaodong Yang, et al. Jarvis-1: Open-world multi-task agents with memory-augmented multimodal language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):1894–1907, 2024.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

Quanting Xie, So Yeon Min, Pengliang Ji, Yue Yang, Tianyi Zhang, Kedi Xu, Aarav Bajaj, Ruslan Salakhutdinov, Matthew Johnson-Roberson, and Yonatan Bisk. Embodied-rag: General non-parametric embodied memory for retrieval and generation. *arXiv preprint arXiv:2409.18313*, 2024.

Yuncong Yang, Han Yang, Jiachen Zhou, Peihao Chen, Hongxin Zhang, Yilun Du, and Chuang Gan. 3d-mem: 3d scene memory for embodied exploration and reasoning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17294–17303. IEEE, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Yuguang Yue, Irakli Salia, Samuel Hunt, Chris Green, Wenzhe Shi, and Jonathan J Hunt. Scaling behavior cloning improves causal reasoning: An open model for real-time video game playing. *arXiv preprint arXiv:2601.04575*, 2026.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

A SUPPLEMENTARY METHOD DETAILS

This section follows the method in the main text: spatial indexing and state conditions (Section 4.2), skill-conditioned outcome compatibility sharing (Section 4.3), and goal-conditioned adaptation and execution (Section 4.4). Experimental implementation and evaluation settings are collected in Appendix B, followed by additional ablations and comparisons in Appendix C.

A.1 SPATIAL INDEXING AND STATE CONDITIONS

The spatial layer stores visually re-identifiable anchors and observed behavioral transitions, not engine pose or global metric coordinates. Observations associate with existing anchors or establish new ones; insufficient association evidence leaves the location unknown. The associated place p and shared non-spatial feature vector z form the experience state, with HUD and auxiliary visual fields defined in Appendix A.2.1. Spatial edges record actual transitions and their quality, not guaranteed stable traversability. SCOCS groups only describe whether records of a particular skill can be pooled, and therefore do not change navigation topology.

Each executed skill segment writes $e = (s, a, \omega, s')$ and retains its supporting RGB, low-level actions, and active instruction. Unexecuted skills create no records. Revisits append samples to the exact state–skill key; grouping never deletes or rewrites original records.

A.1.1 ACTIVE VISUAL LOCALIZATION AND CLOSED-LOOP EXECUTION

Multi-view scanning and place matching. The agent rotates in place, collecting eight views at 45° intervals to form a panorama:

$$\mathcal{P}_t = \{o_t^{(0)}, \dots, o_t^{(7)}\}. \tag{A.1}$$

The current panorama is cyclically matched against the reference panorama of place v_i to accommodate heading changes:

$$\begin{aligned} S(\mathcal{P}_t, \mathcal{P}_i) &= \max_{k \in \{0,\dots,7\}} \frac{1}{8} \sum_{j=0}^{7} s\Big(o_t^{(j)}, o_i^{((j+k) \bmod 8)}\Big), \\ s(o, o') &= 0.75\langle g(o), g(o')\rangle + 0.25\langle c(o), c(o')\rangle. \end{aligned} \tag{A.2}$$

Here g and c are grayscale-structure and color descriptors. Initial candidates undergo panorama registration and multi-view ORB geometric verification. Insufficient evidence or remaining ambiguity returns $\perp$ rather than forcing a match to an existing node.

Starting-condition verification and stepwise correction. By Eq. 3, only edges starting at the current place with an allowed quality status are candidates. Before reuse, current RGB must match the reference view twice consecutively to verify the historical visual starting condition. An unknown location or failed verification does not invoke another place's outgoing edges.

Let k be the historical action cursor and o_k^e its reference frame. ORB correspondences and RANSAC estimate the homography

$$\mathbf{H}_t = \mathrm{RANSAC}\big(\mathrm{ORB}(o_k^e), \mathrm{ORB}(o_t)\big). \tag{A.3}$$

The horizontal deviation Δx_t obtained from $\mathbf{H}_t$ corrects the historical mouse action:

$$\begin{aligned} m_t &= m_k^e + \delta m_t, \\ \delta m_t &= \begin{cases} 0, & |\Delta x_t| < 6, \\ \mathrm{clip}\Big(-\sigma \Delta x_t \cdot 0.10 \frac{0.1}{\eta}, -12, 12\Big), & \text{otherwise.} \end{cases} \end{aligned} \tag{A.4}$$

Here η is mouse sensitivity and σ is the turning sign. Execution follows

$$(u_t, k_{t+1}) = \begin{cases} (u_k^e + \text{turn correction}, k+1), & \text{visual verification passes,} \\ (u_t^{\text{align}}, k), & \text{alignment required,} \\ (u_t^{\text{recover}}, k), & \text{mismatch: exit and relocalize.} \end{cases} \tag{A.5}$$

Standalone alignment or correction actions do not advance the cursor; it advances only after verification of the current reference frame and execution of its historical action. A visual mismatch immediately exits the edge rather than continuing blind replay.

Endpoint verification and node handoff. Completing the action sequence is not sufficient for completing the spatial transition. The cursor must reach the end and endpoint RGB must match twice consecutively; a subsequent Survey must then identify the expected destination:

$$\begin{aligned} \mathrm{EdgeDone} &= \mathbf{1}[k = L]\,\mathbf{1}[\text{two consecutive endpoint RGB matches}], \\ \mathrm{NodeHandoff} &= \mathrm{EdgeDone}\,\mathbf{1}[\text{subsequent Survey identifies expected node}]. \end{aligned} \tag{A.6}$$

These checks separately verify starting conditions, execution progress, and actual arrival, forming an auditable action–observation loop. Current experiments support the navigation benefit of visual memory, but do not attribute completion gains specifically to stable repeated tours or directly establish correction of cross-game semantic misidentification.

A.2 SCOCS Distance, Missing Evidence, and Sharing

A.2.1 Shared Visual State, Base Distance, and Outcome Changes

HUD-derived state. We construct the observable state from information conveyed by each game's HUD and from auxiliary visual observations. In ViZDoom, the HUD fields are current health, held weapon, armor, and ammunition. In Sauerbraten and Xonotic, they are health, armor, ammunition, and kill notifications. These fields are read from RGB using OCR with manually calibrated game-specific settings, rather than engine-state access. Accurate HUD parsing is an observation-interface requirement, not a contribution of our spatial or state memory, so we do not study OCR design in this work. Kill notifications denote visible HUD cues, not privileged kill counters.

Shared auxiliary visual features. We ask Qwen to inspect human demonstration videos and propose visual information useful for decisions. The resulting features include the number of visible enemies, the number of visible resources, whether the agent is under attack, and enemy proximity. At each decision step, Qwen first performs visual grounding question answering (grounding QA) on the current RGB observations to obtain the values of these features, which then inform its decision. These auxiliary features are therefore extracted through Qwen's grounding QA, not through the HUD OCR pipeline. We found that the visual cues proposed by Qwen3.5-9B were largely consistent across the games, and therefore did not introduce additional game-specific feature-design mechanisms. All baseline and ablation configurations share the same HUD reader and non-spatial visual-feature specification. The additional feature supplied by our world index is the associated place, which allows the system to detect movement between regions; it does not provide extra HUD information.

State representation and discretization. Let $\mathcal{F}_g$ index the non-spatial HUD and auxiliary visual fields for game g. Write $z_t = (z_t^f)_{f\in\mathcal{F}_g}$ and $s_t = (p_t, z_t)$, where p_t is the associated world-index place. Health, armor, and ammunition are each discretized into ten ordered levels:

$$z_t^f = q_f(x_t^f) \in \{0, \dots, 9\}, \qquad f \in \{\mathrm{health}, \mathrm{armor}, \mathrm{ammo}\}. \tag{A.7}$$

Here x_t^f is the HUD reading and q_f is the corresponding ten-bin quantizer. Weapon identity and HUD notifications retain their categorical or event meanings. The feature set follows what the game visually exposes; absent or unreadable observations are not replaced by privileged state.

For known non-death states, the base distance compares the shared non-spatial features:

$$d_0(s_i, s_j) = \max_{f\in\mathcal{F}_g} \delta_f^{\mathrm{state}}(z_i^f, z_j^f), \tag{A.8}$$

where $\delta_f^{\mathrm{state}} \in [0, 1]$ denotes a normalized feature discrepancy: numerical features are compared by their scaled differences, and categorical features by mismatch. Two death states have distance 0, and death versus survival has distance 1. Different place IDs alone incur no base-distance penalty, preserving cross-place comparison.

Outcomes as feature changes. For a skill segment with starting and successor states (p, z) and (p', z'), the outcome ω records changes in the observed features, together with whether the agent crosses a world-index region:

$$\omega = (d, \Delta z, \chi_p), \qquad \Delta z = z' \ominus z, \qquad \chi_p = \mathbf{1}[p' \neq p], \tag{A.9}$$

where d indicates death. The operator $\ominus$ takes differences for numerical features, including differences between the ten-bin health, armor, and ammunition levels, and records transitions for categorical or event features rather than subtracting arbitrary category IDs. Thus the outcome captures resource changes, changes in the observed local situation, and inter-region movement. The crossing indicator is unknown when either endpoint place is unknown; raw place IDs are never numerically subtracted.

The immediate outcome discrepancy in Eq. 4 becomes

$$\delta_\omega(\omega_i, \omega_j) = \max\left\{ \mathbf{1}[d_i \neq d_j],\ \max_{f \in \mathcal{F}_g} \delta_f^{\text{change}}(\Delta z_i^f, \Delta z_j^f),\ |\chi_{p,i} - \chi_{p,j}| \right\}, \tag{A.10}$$

where $\delta_f^{\text{change}} \in [0,1]$ compares numerical changes or categorical transitions in the corresponding feature. If only one segment ends in death, the death indicator makes the cost 1. Missing feature changes, including unreadable terminal observations, and unknown region crossings are handled by the distance bounds below rather than treated as zero change. The recursive cost remains $c_h = \max\{\delta_\omega, \gamma d_{h-1}\}$, with depth counted in skill segments rather than game frames.

A.2.2 Missing Observations, Grouping, and Online Parameters

Missing observations are not evidence of similarity. Unknown outcome fields or a successor skill observed on only one side are represented by distance intervals $[\underline{D}_h^a, \overline{D}_h^a]$. The system computes optimistic and pessimistic costs separately and propagates the lower and upper bounds through optimal transport and recursion. Entirely absent comparable evidence gives $[0,1]$. Sharing uses the upper bound $\overline{D}_h^a$ in Eq. 6, rather than treating missing observations as established similarity. These intervals encode missing observations, are neither frequentist nor Bayesian confidence intervals, and do not eliminate small-sample estimation error. Source states need not have identical complete skill sets; for depth greater than 1, however, successor comparisons use the union of observed skills, and the upper bound from one-sided missing evidence may prevent sharing.

Grouping is independent for each skill. Candidates must have associated places, records of the skill on both sides, and matching observable conditions z as defined in Appendix A.2.1. Deterministic greedy complete-link grouping processes candidates in a fixed order: a new state joins a group only if $\overline{D}_h^a \leq \varepsilon$ holds against every member. This is not a global optimization, so processing order can affect the partition.

Online implementation fixes $\gamma = 0.9$ and $\varepsilon = 0.15$, and constructs sharing groups at $h = 2$ and $h = 1$. Retrieval fallback order, returned summary count, and token budget are given in Appendix A.2.3.

A.2.3 Sharing-Group Updates and Queries

Sharing groups are recomputed or incrementally updated from the current experience library. Only states with associated places and samples of the target skill on both sides become candidates; source states need not have identical complete skill sets. Recursive successor comparisons and missing evidence are described in Appendix A.2.2.

At a decision event, visual relocalization produces place estimate $\hat{p}_t$ and association confidence ρ_t. The shared HUD and auxiliary visual features form z_t, giving the exact query key $s_t = (\hat{p}_t, z_t)$. The system tries, in order, the depth-2 skill-sharing group containing this state, its depth-1 group, and unshared records at the current place. An unknown place or absent exact state key does not borrow across places.

Group records are directly pooled as a sample multiset rather than first compressed to means:

$$\mathcal{D}(G_a, a) = \biguplus_{s \in G_a} \mathcal{D}(s, a). \tag{A.11}$$

Each summary reports sample count, number of source places, current-place support, observed HUD and auxiliary-feature changes, inter-region movement, death count, and duration range. A query returns at most three summaries within a total budget of 768 tokens. Let τ_k denote a VLM decision event, r the RGB window lookback, and $\mathcal{C}_k$ the retrieved summaries. Skill selection is

$$a_k \sim \pi_{\text{VLM}}\big(a \mid o_{\tau_k - r:\tau_k}, x, s_{\tau_k}, \mathcal{C}_k, \mathcal{M}_{\text{local}}\big). \tag{A.12}$$

The VLM treats summaries as historical evidence and decides whether to use them based on current RGB, task, and local spatial context, rather than letting group membership directly choose an action.

A.3 Visual–Action Adaptation and Goal-Conditioned Execution

The following expands the downstream adaptation and execution in Section 4.4, distinguishing standalone visual–action control from goal-conditioned control in GameWISE-Hybrid. Neither variant changes the separation of spatial identity from skill-experience applicability or participates in the SCOCS sharing criterion.

Standalone visual–action adaptation. GameWISE-VLA uses only observation–action pairs $(o_{t-r:t}, u_t)$ from the collected demonstrations for behavior-cloning adaptation, without skill or grounding inputs. At evaluation, it predicts low-level actions from visual observations, with no task-language input, skill selection, target slots, image-coordinate conditions, or online VLM interaction. Its cross-engine evaluation therefore tests transfer of behavior learned from visual–action demonstrations, not instruction following or online VLM guidance. The goal-conditioned interface and persistent execution described below apply to the controller in GameWISE-Hybrid, not to standalone GameWISE-VLA.

Supervision and persistent execution in the hybrid. Every executed segment is associated with active local instruction ℓ_i, goal condition c_t, and observed outcomes. Screening is skill-local: it checks goal-behavior completion, spatial-association consistency, and target-label validity. Failures remain in the SCOCS library, but only segments passing screening enter $\mathcal{D}_{\text{adapt}}$. Sampling preserves coverage across valid skills and targets, preventing frequent behaviors from dominating adaptation.

Training extracts $(o_{t-r:t}, c_t, u_t)$ from retained segments and optimizes the goal-conditioned behavior-cloning losses in Eqs. A.27 and A.29. At deployment, the VLM selects a skill and local instruction ℓ_k at τ_k and grounds them into target slots, image coordinates, and validity indicators c_{τ_k}. Between decisions, the VLA continues acting from current vision:

$$u_t \sim \pi_\theta(u_t \mid o_{t-r:t}, c_t), \qquad \tau_k \le t < \tau_{k+1}. \tag{A.13}$$

Thus the persistent object is a local goal, not a fixed action sequence. When visual tracking is enabled, the target point in c_t changes with the image. Skill completion, condition invalidation, target changes, or relevant events trigger replanning. The VLA does not directly consume free-form ℓ_k, but executes it through grounded conditions. Online interaction can update $\mathcal{M}$, the experience library, and skill-sharing groups, while the current training procedure performs only one offline adaptation pass.

From Internet behavior priors to goal-conditioned control. Let Internet visual–action data be

$$\mathcal{D}_{\text{VA}} = \{(o_{1:T}, u_{1:T})\}.$$

Pretraining learns visually conditioned human operation through behavior cloning:

$$\theta_0 = \arg\min_\theta \mathbb{E}_{\mathcal{D}_{\text{VA}}} \left[-\sum_t \log \pi_\theta(u_t \mid h_t) \right], \tag{A.14}$$

where h_t is the observation–action context available to the policy. However,

$$\pi_\theta(u_t \mid h_t) \quad \not\Rightarrow \quad \pi_\theta(u_t \mid h_t, \text{desired goal}).$$

The same image may admit exploration, retreat, or attack. Without goal supervision aligned with actions, the policy cannot explicitly distinguish these intentions. Small models also lack the capacity and context budget for complete long-horizon decision-making, while continuously generating low-level actions with a large VLM incurs high latency. We therefore retain pretrained operation skills and supplement them with executable goal conditions from memory-supported interaction.

Converting interaction memory into adaptation data. At decision event t_k, the VLM uses the current observation, spatial memory $\mathcal{M}_k$, and skill-experience structure $\mathcal{G}_k$ to select a skill, generate a local instruction, and locate an image target:

$$(a_k, \ell_k, g_{t_k}) \sim \Pi_{\text{VLM}}(o_{\le t_k}, \mathcal{M}_k, \mathcal{G}_k). \tag{A.15}$$

A skill attempt forms an intention-labeled interaction segment:

$$\xi_k = \left(v_k, a_k, \ell_k, \{o_t, g_t, u_t\}_{t=t_k}^{t_{k+1}}, \omega_k\right), \tag{A.16}$$

where v_k is the spatial anchor, a_k the skill, ℓ_k its semantic instruction, g_t the image target, u_t the low-level action, and ω_k the skill outcome. Segments are screened for spatial consistency, verified outcomes, and valid target labels:

$$w_k = \mathbf{1}[\mathrm{Success}(\xi_k)]\mathbf{1}[\mathrm{SpatialConsistent}(\xi_k)]\mathbf{1}[\mathrm{GoalValid}(\xi_k)], \tag{A.17}$$

$$\mathcal{D}_{\mathrm{adapt}} = \{\xi_k : w_k = 1\}. \tag{A.18}$$

Equation A.17 formalizes screening; it does not imply three identically named functions in the implementation. Online interaction produces both environmental experience and goal-conditioned supervision. SCOCS retains successful and failed records for comparing outcomes, whereas VLA adaptation uses only screened locally valid demonstrations. What transfers to the VLA is successful behavior generated by memory-supported decisions, not the spatial or experience graph data structure.

Compact skill–target interface. Before execution, the VLM's semantic intention is compressed into target-slot type κ_t, normalized image coordinates g_t, and a validity indicator b_t:

$$c_t = (\kappa_t, g_t, b_t), \qquad g_t = (x_t, y_t) \in [0,1]^2, \qquad \kappa_t \in \{\mathrm{target}, \mathrm{nav}\}. \tag{A.19}$$

`VlaExploreExecutor` uses the `nav` slot for navigation, and `VlaShootExecutor` uses the `target` slot for movement, aiming, and shooting. Local instructions are thus grounded through current RGB into compact control conditions:

$$\ell_k \xrightarrow[\text{current RGB}]{\text{VLM}} (\kappa_t, g_t) \xrightarrow{\text{VLA}} u_t. \tag{A.20}$$

Skill type expresses what to do; the target point indicates where to act. A target point is a local handoff interface, not a complete task description.

Dual encoding of target coordinates and target-relevant visual content. The hybrid harness defaults to a Qwen3.5-0.8B per-frame `affordance-concat` policy using ViT layer-12 features, loaded through `QwenStage3Lightning`. The model receives 512×512 RGB. For each target slot $\kappa \in \{\mathrm{target}, \mathrm{nav}\}$, the normalized point g_t^κ is mapped to pixel coordinates (u_κ, v_κ) and converted into a Gaussian heatmap using the slot's validity indicator b_t^κ:

$$H_t^\kappa(u,v) = b_t^\kappa \exp\left(-\frac{(u-u_\kappa)^2 + (v-v_\kappa)^2}{2\sigma^2}\right), \qquad \sigma = 16. \tag{A.21}$$

The visual encoder extracts layer-12 RGB features:

$$X_t = E_{\mathrm{vision}}^{(12)}(o_t). \tag{A.22}$$

The heatmap is resized to the visual-token grid and converted into an attention bias:

$$B_t^\kappa = 2\log\left(\max(\mathrm{Resize}(H_t^\kappa), 0.05)\right). \tag{A.23}$$

The core target-relevant feature computation is

$$F_t^\kappa = \mathrm{softmax}\left(\frac{Q_\kappa K_t^\top}{\sqrt{d}} + B_t^\kappa\right) V_t, \qquad K_t = X_t W_K, \quad V_t = X_t W_V. \tag{A.24}$$

The actual module also includes multi-head attention, normalization, residual connections, and feed-forward layers. Each slot uses four focus queries; `target` and `nav` together produce eight target-visual tokens. Coordinates and validity are also encoded by an MLP with slot-identity and positional embeddings:

$$C_t^\kappa = f_{\mathrm{coord}}(g_t^\kappa, b_t^\kappa) + e_\kappa + e_{\mathrm{pos}}. \tag{A.25}$$

The final input fuses the original visual tokens, target-visual tokens from both slots, and coordinate tokens:

$$Z_t = \mathrm{Concat}\left(X_t, F_t^{\mathrm{target}}, F_t^{\mathrm{nav}}, C_t^{\mathrm{target}}, C_t^{\mathrm{nav}}\right). \tag{A.26}$$

The coordinate branch explicitly specifies target position, while the focus branch reads actual visual content near that target. Gaussian heatmaps bias attention over RGB tokens rather than being concatenated as additional RGB channels.

Goal-conditioned adaptation and action decoding. Starting from pretrained visual–action parameters θ_0, the policy learns goal-conditioned control from screened demonstrations:

$$\theta^\star = \arg\min_{\theta \leftarrow \theta_0} \mathbb{E}_{\xi \sim \mathcal{D}_{\text{adapt}}} \left[-\sum_t \log \pi_\theta(u_t \mid h_t, c_t) \right]. \tag{A.27}$$

This objective provides two types of supervision:

$$\underbrace{c_t \to u_t}_{\text{follow VLM-grounded goals}} \qquad \underbrace{(o_t, c_t) \to u_t}_{\text{adapt to target-environment control}}.$$

A three-layer autoregressive action decoder predicts keyboard, mouse buttons, and discretized mouse displacement in a fixed order:

$$\pi_\theta(u_t \mid h_t, c_t) = \prod_j p_\theta \left(u_t^{(j)} \mid h_t, c_t, u_t^{(<j)} \right). \tag{A.28}$$

The training loss is channel-wise cross-entropy normalized by action-space size plus z-loss, not KL divergence from teacher logits:

$$\mathcal{L} = \sum_j \lambda_j \frac{\text{CE}_j}{\log |\mathcal{A}_j|} + 10^{-4} \sum_j \lambda_j \mathcal{L}_{z,j}, \tag{A.29}$$

with current channel weights

$$(\lambda_{\text{key}}, \lambda_{\text{button}}, \lambda_{\Delta x}, \lambda_{\Delta y}) = (1, 1, 1, 0).$$

VLM decisions and high-frequency VLA execution. At deployment, the VLM retains skill selection and target generation, while the adapted VLA takes over low-level actions:

$$(a_k, g_{t_k}) \sim \Pi_{\text{VLM}}(o_{\le t_k}, \mathcal{M}, \mathcal{G}), \tag{A.30}$$

$$u_t \sim \pi_{\theta^\star}(u_t \mid h_t, a_k, g_t), \qquad t_k \le t < t_{k+1}. \tag{A.31}$$

The VLA continues the local skill between VLM updates. When visual tracking is enabled, the target point follows the image:

$$g_t = \text{Track}(o_t, g_{t-1}). \tag{A.32}$$

A.4 Structured Messages and Experience Records

Decision-time context contains the current RGB window, task, exact state key, matched spatial neighborhood and frontiers, candidate skills, and SCOCS summaries with sample-support information. The VLM outputs selected skill a_k, semantic instruction ℓ_k, target slots and image coordinates, and completion or replanning conditions. During hybrid execution, the VLA receives the current RGB window and compact goal condition c_t, not the spatial or experience graph itself. At segment completion, changes in the shared HUD and auxiliary visual features, the inter-region movement indicator, death, segment duration, successor state, and the visual–action trajectory are written to the library. Success, spatial consistency, and target validity determine inclusion in $\mathcal{D}_{\text{adapt}}$.

B Experimental Setup

B.1 Implementation Details

VLM and executor. All VLM-based methods use a frozen Qwen3.5-9B model deployed locally at temperature 0, without task-specific fine-tuning. Only the VLM component of GameWISE-Hybrid issues a directive or grounding request every 16 environment steps. All other VLM agents, including GameWISE-VLM and every VLM baseline, issue such requests every 4 environment steps. In both cases, the agent blocks on the response, with a 60 s timeout; the waiting time is charged to the requesting step, so the reported p95 step latency includes planner waiting time rather than measuring planner inference alone. The planner routes only among the Explore, Pickup, and Combat executors. Standalone GameWISE-VLA predicts keyboard–mouse actions from RGB without online VLM calls, skill selection, grounding, or task-language inputs. In contrast, GameWISE-Hybrid uses its VLM to select skills and ground local goals for the goal-conditioned controller. All systems use the same low-level keyboard–mouse action space.

Baseline configurations. All VLM memory baselines and memory ablations receive the same non-spatial HUD and auxiliary visual features described in Appendix A.2.1. The memoryless control routes from the current visual observation and these features without retrieving history. SPTM (Savinov et al., 2018) serves as the topological spatial-memory baseline within the shared VLM–executor harness; its reported results refer to this common-harness configuration. ReAct + Reflexion injects a thought–action–outcome trace into the prompt and persists an end-of-episode reflection into subsequent prompts. ExpeL contrasts successful and failed episodes at episode end and maintains an insight rule list through planner-issued add and delete operations. RAP retrieves past decision blocks by situation-feature similarity and supplies them as few-shot context. JARVIS-1 extracts positive and negative (situation, plan) pairs at episode end and prefers positive matches at retrieval. End-of-episode memory-writing requests are scheduled at higher priority than directive and grounding requests and are exempt from the goal-revision check, so that every reflection or insight commit reported here is an actual planner output rather than a template fallback. Thus the comparison among the VLM baselines and GAMEWISE-VLM controls the planner, four-step decision interval, rule-based executors, and interaction budget. The hybrid comparison instead evaluates the combined system with a sixteen-step VLM interval and VLA execution; it is not a matched-frequency memory-only comparison.

Latency interpretation. The hybrid uses a four-times-longer scheduled planner interval (16 versus 4 environment steps), so planner calls are less frequent per environment step. Together with VLA execution between calls, this changes the distribution of step latencies. A lower p95 step latency does not imply faster individual VLM inference, nor does p95 necessarily scale in proportion to call frequency. The reported aggregate latency does not separate planning, execution, retrieval, and memory-update costs, so it cannot attribute the observed reduction to any one component.

VLA training and baselines. Our policy adopts the Qwen3.5-0.8B architecture and adds action tokens and an action decoder. OpenP2P and NitroGen are initialized from their officially released pretrained checkpoints, whereas our policy is pretrained from scratch for 50,000 steps with batch size 8 on the public OpenP2P dataset, which contains Doom and ViZDoom gameplay; all policies are then post-trained on the same interaction data collected by our deployed agent. Standalone GAMEWISE-VLA uses only the visual–action demonstration pairs for this adaptation, without skill or grounding inputs; goal-conditioned adaptation for GAMEWISE-Hybrid additionally uses the target annotations described in Appendix A.3. Because the pretraining corpus already contains Doom-family data, the ViZDoom results of the VLA policies reflect adaptation from source-game interaction rather than zero-shot generalization; no Xonotic or Sauerbraten trajectory is used for policy pretraining or adaptation. This training-data exclusion is distinct from the manually calibrated HUD reader and demonstration-informed visual-feature specification, which define the shared observation interface. Each policy retains the head-specific configuration from its open-source implementation, and evaluation uses the final checkpoint. Training requires approximately 90 GB of GPU memory.

Runtime and action interface. During interaction, RGB observations update the spatial graph and supply the shared HUD and auxiliary visual features, forming the state key (p, z). Spatial frontiers guide expansion, and executed skill segments add feature-change outcomes and update the corresponding sharing groups. At hybrid deployment, the VLM reads the current visual state, local spatial memory, and SCOCS summaries at its decision frequency. Its grounded instruction persists while the VLA emits keyboard, mouse-button, and mouse-displacement actions at the environment control frequency. Instruction completion, execution failure, target changes, or relevant events trigger new VLM decisions.

The hybrid controller receives RGB history and condition c_t, comprising target slots, image coordinates, and validity indicators, and uses Eqs. A.27 and A.29 for goal-conditioned adaptation. Standalone GAMEWISE-VLA omits these goal conditions and uses visual–action demonstrations only. Training uses screened segments in $\mathcal{D}_{\mathrm{adapt}}$; unexecuted actions are not converted into labels. The SCOCS state key includes the HUD and auxiliary visual fields specified in Appendix A.2.1; terrain semantics and task progress may additionally appear in the VLM's RGB context. The current procedure ends after this offline adaptation stage and does not conduct a second exploration round using the adapted VLA.

B.2 Evaluation Protocol

B.2.1 Tasks and Runs

ViZDoom My Way Home evaluates route retrieval, Deathmatch evaluates resource- and state-dependent combat, and Defend the Center provides a limited-navigation control. Xonotic and Sauerbraten test transfer under rendering and control shifts, and additionally test whether memory can be acquired and grown online in an unfamiliar engine. Each cell is evaluated for 20 episodes with seeds 700–719, all environments run at 20 FPS, and episode limits are 4200 tics for Deathmatch and 2100 tics for My Way Home and Defend the Center. All reported numbers are per-cell means. Table B.1 summarizes the tasks and metrics.

Table B.1: Evaluation suite. The reported metrics are exactly those appearing in Tables 1 and 2. In addition to the per-scenario metrics listed here, p95 step latency is reported for every engine. All deployed agents observe RGB only, without engine state; task text is supplied to VLM planners, whereas standalone GameWISE-VLA uses no task-language, skill, or grounding inputs.

Engine	Scenario	Long-horizon requirement	Reported metrics
ViZDoom	Deathmatch	Explore, acquire resources, revisit useful regions, fight, and recover after respawn	cumulative reward, kills
ViZDoom	My Way Home	Navigate an unseen maze to the goal without combat	success rate, SPL
ViZDoom	Defend the Center	Visually track approaching enemies while remaining near the center	cumulative reward (scenario scoring over kills and deaths)
Xonotic	held-out deathmatch maps	Online memory acquisition plus transfer of exploration and combat	kills, coverage, travel distance
Sauerbraten	held-out deathmatch maps	Online memory acquisition under new rendering and dynamics	kills, coverage, travel distance

B.2.2 My Way Home Metrics

The scenario grants $+1$ on reaching the goal object and a living reward of -10^{-4} per tic, so episode return is $1 - 10^{-4}T_{\text{goal}}$ on success and exactly -0.21 on a full 2100-tic timeout. Return is therefore a deterministic function of success and time-to-goal, and its floor reflects failure to reach the goal rather than a penalty. We consequently report success rate and SPL. Success requires ending within 64 units of the goal, and

$$\text{SPL} = \frac{1}{N}\sum_{i=1}^{N} S_i \frac{\ell_i}{\max(p_i, \ell_i)}, \tag{B.1}$$

where ℓ_i is the A* shortest-path length from the spawn point and p_i is the distance actually travelled. SPL upper-bounds nothing on its own: it equals the success rate times the mean path efficiency on successful episodes, so the two metrics separate whether an agent reaches the goal from whether it takes a direct route.

B.2.3 Interaction and Reuse

Each test environment starts with empty memory and receives a fixed interaction budget before evaluation on held-out approach views and changed task or resource conditions. Frozen-memory evaluation measures reuse, while online updates additionally measure acquisition. Memory variants are constructed from a shared recorded history before their online trajectories diverge; interaction steps, memory and token budgets, VLM, skills, and action mappings are matched. Memory is never transferred between test environments: the VLM scaffold and the hybrid system rebuild it online from empty memory in every engine, including the held-out ones, whereas the adapted VLA policies carry only the parametric policy and require no target-engine memory. Engine state is used only for offline labels and metrics.

B.2.4 CROSS-ENGINE SCOPE

Cross-engine checkpoints receive no target-engine fine-tuning, and no policy is trained on any Xonotic or Sauerbraten trajectory. Observation resizing and action mappings are fixed before evaluation. GAMEWISE-VLM and GAMEWISE-Hybrid start from empty memory and rebuild it online in each held-out engine under the same fixed-budget interaction protocol used in ViZDoom, so the cross-engine table measures whether the pipeline can acquire and grow spatially anchored experience in an unfamiliar environment, in addition to whether the adapted parametric policy transfers. The online VLM scaffold is evaluated through the same RGB-only interface as the learned policies. Specifically, standalone GAMEWISE-VLA is adapted using only the visual–action demonstration pairs from experience accumulated by GAMEWISE-VLM over 20 source-game collection episodes. It is then evaluated on held-out engines without skill or grounding inputs, online VLM interaction, or further policy updates. GAMEWISE-Hybrid uses the goal-conditioned controller described in Appendix A.3 for a fresh evaluation, with the VLM's memory cleared before testing and rebuilt online.

B.2.5 DEATHMATCH STATISTICAL TESTING

All methods are evaluated on the same seeds (700–719), so scenario-level comparisons are made on within-seed differences rather than on marginal dispersion. For each metric we compute $d_i = x_i^{\text{method}} - x_i^{\text{ref}}$ over the 20 shared seeds, taking the memoryless control as the reference, and report the mean difference, a percentile bootstrap 95% CI ($10{,}000$ resamples), the Hodges–Lehmann median difference, and a two-sided Wilcoxon signed-rank p value with Pratt zero handling. Bold entries in Table 1 mark the numerically best value per column and are not significance claims.

C ADDITIONAL ABLATIONS AND COMPARISONS

This section supplements the ablations and analysis in Section 5.4 and the system-level comparison in Table 1.

C.1 SCOCS SENSITIVITY AND COMPONENT ABLATIONS

The online values $\gamma = 0.9$, $\varepsilon = 0.15$, and $h \in \{1, 2\}$ are design choices rather than measured optima. This subsection reports how selectivity and behavior move with these parameters, and attributes the observed gain to individual parts of the sharing criterion.

C.1.1 PROTOCOL

All arms share one frozen experience library, spatial index, planner, executor set, and 768-token retrieval budget; Each arm is then run on fresh episodes with evaluation seeds 700–719, which do not overlap the collection seeds, for two rounds per seed at VLM temperature 0 under strict blocking. **The two rounds of each seed are first averaged, so the paired unit is a seed and $n = 20$ throughout**; we do not treat the 40 runs as independent observations. Offline quantities are computed once per parameter cell, since the recursive distances depend only on (γ, h) and re-thresholding ε within a fixed (γ, h) is free. Rewards in this subsection correspond to the GAMEWISE-VLM configuration, that is, spatial indexing and SCOCS without the persistent execution interface. Throughout, a *group* is a non-singleton skill-sharing group at the stated depth, and *support* counts the records those groups make consultable; depths are reported separately because deployment uses the $h{=}2 \to h{=}1$ fallback.

C.1.2 SHARING-THRESHOLD SENSITIVITY

Table C.1: Sharing-threshold sensitivity at $\gamma = 0.9$ on ViZDoom Deathmatch. Groups and support are offline quantities at each depth; reward is the per-seed mean with its across-seed standard deviation ($n = 20$ seeds, two rounds averaged within each seed). The $\varepsilon = 1.00$ cell is equivalent to unconditional pooling within the root-state gate, and is the same arm as `pool` in Table C.2 rather than independent evidence from it.

	Groups		Support records		
ε	h=1	h=2	h=1	h=2	DM reward ↑
0.05	425	353	6762	5271	9.05 ± 3.45
0.10	358	362	7043	5480	12.10 ± 3.07
0.15 (deployed)	326	369	7106	5570	$\mathbf{13.24 \pm 3.41}$
0.30	281	385	7232	5847	13.12 ± 3.33
0.50	270	402	7447	6188	12.03 ± 3.01
1.00 (≈ pooling)	135	135	7826	7826	9.00 ± 2.27

Table C.1 shows that ε acts as a smooth selectivity control rather than a single operating point. Relaxing it monotonically reduces the number of depth-1 groups ($425 \rightarrow 270$) while enlarging them and increasing support ($6762 \rightarrow 7447$), continuously approaching unconditional pooling. Reward is unimodal, and $\varepsilon = 0.15$ and $\varepsilon = 0.30$ are indistinguishable (13.24 versus 13.12): the deployed value lies inside a plateau rather than at a sharp peak, while both ends of the grid lose 3–4 reward. Depth-2 group counts move in the opposite direction from depth-1 counts because a stricter criterion fragments borderline groups into singletons, which this count excludes by definition; support at depth 2 is the quantity that tracks permissiveness monotonically.

C.1.3 SCOCS COMPONENT ATTRIBUTION

Table C.2: Seed-paired SCOCS criterion component attribution on ViZDoom Deathmatch under GAMEWISE-VLM. The two rounds of each seed are first averaged, so the paired unit is a seed and $n = 20$. Δ is the mean within-seed difference against the full criterion, and the interval is the corresponding 95% paired t-interval ($t_{0.975,19} = 2.093$). `pool` is the $\varepsilon = 1.00$ cell of Table C.1.

Variant	Reward ↑	Δ vs. full (95% CI)
SCOCS (full)	**13.24**	–
Unconditional pooling (`pool`, ε=1.00)	9.00	-4.24 ± 1.58
Immediate only (γ=0)	10.12	-3.12 ± 1.82
Optimistic-bound grouping (`no_upper`)	8.38	-4.86 ± 2.43

$\mathrm{SD}_d = 3.37$, 3.89, and 5.20 for the three ablations in row order. The repeated SCOCS mean is a common reference, not independent evidence; it is the 13.24 VLM setting, not the 15.33 full system.

Table C.2 evaluates three changes to the sharing criterion. Unconditional pooling within the root-state gate (`pool`), removing recursive successor comparison ($\gamma = 0$), and grouping on the optimistic rather than the conservative bound (`no_upper`) each lower mean reward relative to the full criterion, with paired 95% confidence intervals for the differences excluding zero. The largest observed mean decrease is 4.86 for `no_upper`, supporting conservative rather than optimistic treatment of missing evidence in this setting. These comparisons do not establish the mechanism underlying the performance differences.

C.2 SEED-PAIRED COMPARISON WITH MEMORY BASELINES

Table C.3 supplements the Deathmatch comparison in Table 1, using the seed-paired testing protocol in Appendix B.2.5.

Table C.3: Seed-paired differences between prompt-level memory mechanisms and the memoryless control on ViZDoom Deathmatch ($n = 20$, seeds 700–719). Δ is the mean within-seed difference, CI a percentile bootstrap 95% interval over $10{,}000$ resamples, HL the Hodges–Lehmann median difference, and p a two-sided Wilcoxon signed-rank test with Pratt zero handling. No prompt-level mechanism separates from the control on this scenario, whereas GAMEWISE-VLM separates on both reward and kills.

Method	Mean	Δ	95% CI	HL	p
Reward					
No memory (reference)	5.10	–	–	–	–
ReAct + Reflexion	2.40	-2.70	$[-5.90, -0.10]$	-1.00	0.206
SPTM	7.40	$+2.30$	$[-1.30, +5.95]$	$+2.50$	0.141
ExpeL	5.10	$+0.00$	$[-3.45, +3.95]$	0.00	0.954
RAP	6.30	$+1.20$	$[-3.80, +7.20]$	-1.00	0.430
JARVIS-1	7.10	$+2.00$	$[-1.70, +5.65]$	$+1.50$	0.336
GAMEWISE-VLM (Ours)	13.24	$\mathbf{+8.14}$	$[+4.40, +17.20]$	$+8.50$	$\mathbf{0.0038}$
Kills					
No memory (reference)	2.05	–	–	–	–
ReAct + Reflexion	1.15	-0.90	$[-2.30, +0.20]$	0.00	0.553
SPTM	3.45	$+1.40$	$[-0.10, +2.95]$	$+1.25$	0.093
ExpeL	2.15	$+0.10$	$[-1.35, +1.75]$	0.00	0.789
RAP	2.70	$+0.65$	$[-1.45, +3.05]$	0.00	0.775
JARVIS-1	3.10	$+1.05$	$[-0.70, +2.90]$	$+1.00$	0.257
GAMEWISE-VLM (Ours)	5.75	$\mathbf{+3.70}$	$[+2.05, +7.10]$	$+3.50$	$\mathbf{0.0028}$

The uploaded supplementary material includes demonstration videos of GAMEWISE in all three games: ViZDoom, Xonotic, and Sauerbraten.