Red Alert Bench: When LLM Agents Play Differently on the Same Game

Anonymous ACL submission

Abstract

LLM agents are increasingly evaluated on games, yet the headline metric is almost always a single win rate. We show that two agents winning the same fraction of episodes can be playing fundamentally different games, and aggregate scores obscure the behavioural differences researchers most want to study. We introduce Red Alert Bench, a diagnostic benchmark built on a deterministic reimplementation of the real-time strategy game *Red Alert*, with 218 human-authored scenarios across 11 capability families and a symmetric self-play setting. Each scenario targets a single capability and is validated to require that capability and reject shortcut strategies, so performance is attributable rather than hacked. We evaluate eight contemporary LLMs across more than 15,500 episodes, anchored by 77 human-tester episodes. Because the harness records every observation, utterance, and command, we surface behavioural signatures aggregate win rate hides: a model whose ranking inverts between solo and adversarial play, a spawn-side asymmetry that distorts head-to-head comparisons, a fivefold spread in passivity (where the most patient agent, the human cohort, wins most often), and scenarios designed to test one capability that systematically fail along an adjacent one. We release the engine, benchmark, harness, all per-episode artefacts, and a live leaderboard.

1 Introduction

Many benchmarks for large language model (LLM) agents report a single pass/fail signal: a task-suite success rate, a leaderboard win count, an Elo from pairwise comparisons. These summaries rank agents but say little about *how* they succeed or fail: two agents with the same win rate may pursue coherent but different strategies, break down at different stages of the decision pipeline, or respond to setbacks in opposite ways. The same gap appears in the recent wave of gaming benchmarks for LLM agents, which exercise rich behaviours but still distil them into one number.

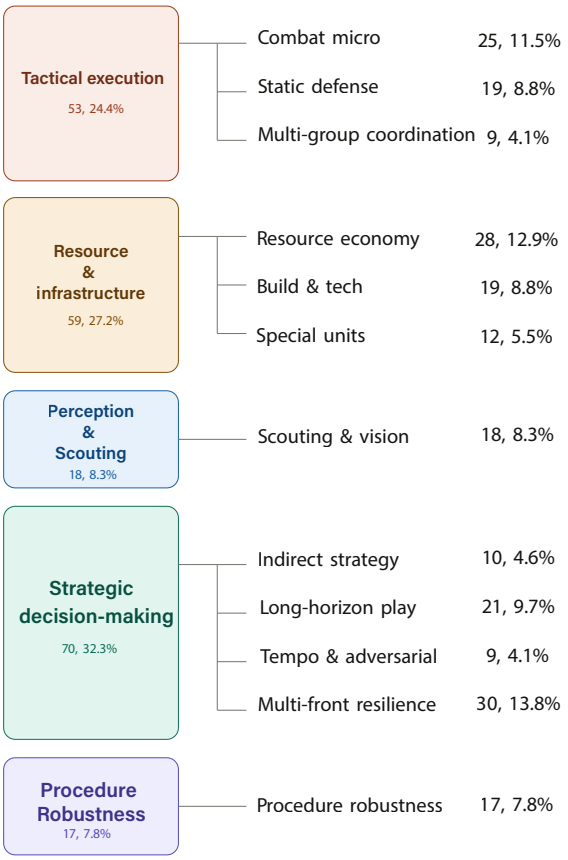

Figure 1: The Red Alert Bench v1.1 capability taxonomy and scenario distribution. The 218 scenario is partitioned into 11 families across four superordinate categories spanning the tactical, economic, perceptual, and strategic axes of RTS play. Three orthogonal labels *perception*, *reasoning*, *action*, along with an *adversarial* setting, project the families onto the perception–reasoning–action chain.

We introduce Red Alert Bench, a benchmark for LLM agents in a real-time strategy (RTS) setting designed to make this behavioural record a first-class object of study. It has three components: a *human-designed scenario suite* of 218 human-authored scenarios across 11 capability families (combat, econ-

omy, scouting under fog of war, base building, defence, coordination, indirect strategy, long-horizon play, tempo, special units, multi-front resilience), each at three difficulty levels and tagged with a single intended capability; a *deterministic, open-source game engine*, built on a reimplementation of the 1996 RTS game, *Red Alert*, that records every observation, utterance, and command together with fine-grained progress on each component of the win condition, and admits a scenario only if trivial shortcut strategies demonstrably fail; and an *evaluation and analysis pipeline* that runs each scenario, varies the agent's perception channel (pure text, labelled minimap composed of text and images, or pure image), injects scripted opening moves to probe recovery, and pits agents against each other in symmetric self-play.

We evaluate eight contemporary LLMs across roughly 15,500 scenario episodes and anchor the leaderboard with 77 human episodes. Our per-episode analysis uncovers behaviours invisible at the win-rate level: a mid-ranked solo model leading the adversarial cohort by committing to a focused early attack; a top solo model under-performing in self-play because it spends its opening shopping for too much technology; a substantial spawn-side asymmetry on the canonical self-play map; a five-fold spread in how often agents choose to wait rather than act, with humans the most patient and also the most successful; scenarios designed to test one capability that systematically fail along an adjacent one; and per-scenario decision instability that persists even when mean win rate is highly stable.

Our contributions are:

- A human-designed scenario suite of 218 hand-authored RTS scenarios across 11 capability families, each at three difficulty levels, with automated validation against trivial shortcut strategies.
- A deterministic, open-source RTS engine that exposes a per-turn LLM-agent interface, records the full observation/utterance/-command trace and fine-grained objective progress, and supports a perception ablation, a scripted-prefix recovery setting, and a symmetric self-play battleground.
- An evaluation of eight contemporary LLMs across roughly 15,500 scenario episodes and 200 head-to-head matches, anchored by 77 human-tester episodes, with a behavioural analysis that aggregate win rate would not have surfaced.
- A public release of the engine, the benchmark, the evaluation harness, all per-episode artefacts, and a live leaderboard.

2 Related Work

LLMs in games. Prior LLM evaluations on RTS games target StarCraft. TextStarCraft II (Ma et al., 2024) wraps StarCraft II behind a text interface and ranks agents on win rate against built-in AI; SwarmBrain (Shao et al., 2024) layers reflex and deliberation modules over the same game; LLM-PySC2 (Li et al., 2024) reports that reasoning ability does not directly translate to decision quality. These benchmarks report one number per game. Red Alert Bench instead decomposes play into 218 scenario packs spanning eleven capability families, each with a deterministic no-cheat bar, so that a single win rate can be split into the perception, reasoning, and action behaviours that drive it. The broader LLM-game literature covers turn-based 4X (CivRealm (Qi et al., 2024), CivBench (Chen et al., 2026)), Pokemon (Karten et al., 2026), interactive fiction (TextQuests (Phan et al., 2025)), pixel-input retro games (VideoGameBench (Zhang et al., 2025)), Diplomacy (Meta Fundamental AI Research Diplomacy Team (FAIR) et al., 2022; Wongkamjan et al., 2024), and cross-game meta-suites (BALROG (Paglieri et al., 2024), SmartPlay (Wu et al., 2024), GameBench (Costarelli et al., 2024), Orak (Lim et al., 2025)). All of these pause the world while the agent thinks. RTS is the harder regime: the simulation keeps advancing, fog of war hides most of the map, and a win requires a long-horizon sequence of dependent decisions under an adversary.

Planning and embodied AI. PlanBench (Valmeekam et al., 2023a) and ALFWorld (Shridhar et al., 2021) probe symbolic planning in static or fully-observed settings; iVISPAR (Mayer et al., 2025) extends planning to a vision-language interactive setup. Vision-language-action systems (OpenVLA (Kim et al., 2024), π_0 (Black et al., 2024), VoxPoser (Huang et al., 2023)) ground a perceive-reason-act loop in robotic manipulation, and EmbodiedBench (Yang et al., 2025) surveys the resulting evaluations. Philion et al. (2020) establish a perception-to-planning metric in autonomous driving. Red Alert Bench

shares this perceive-reason-act framing but moves it to an adversarial, real-time setting where the opponent and the clock both push back on the agent's plan.

Agent evaluation and behavioural analysis. Tool-use agent benchmarks (GAIA (Mialon et al., 2023), WebArena (Zhou et al., 2024), AgentBench (Liu et al., 2024), SWE-bench (Jimenez et al., 2024), τ-bench (Yao et al., 2024), τ^2-bench (Barres et al., 2025)) score agents on a single outcome metric per task; τ-bench's pass^k adds a reliability axis that we adopt. None of these settings has an opponent or partial observability. A growing line of work argues that outcome metrics alone obscure how an agent reaches the outcome: Gao et al. (2026) propose a taxonomy of agent actions and sub-actions and show that behavioural-level analysis surfaces failure modes that aggregate scores hide; Wongkamjan et al. (2024) make the same point in Diplomacy, where CICERO's headline win rate masks a cooperation deficit visible only in per-turn traces. Our findings (Section 5) operationalise this view at bench scale: per-turn artefacts surface opening-template lock-in, passivity, and a cross-tagged perception/reasoning entanglement that the headline win rate alone would not have surfaced.

3 The Red Alert Bench

3.1 Design Highlights

Red Alert Bench is built on a deterministic Rust reimplementation of *Red Alert* (Westwood, 1996), a 2D real-time strategy game in which each player builds a base, harvests resources to fund a production chain, and either eliminates the opponent's base or satisfies a scenario-specific objective under a turn budget.[1] Most LLM-agent benchmarks evaluate in settings that strip out the conditions that make agent behaviour most informative: the world pauses while the agent thinks, observations are fully known, opponents are absent, and 'success' is whatever the rubric is easiest to score. Red Alert Bench puts those conditions back. Four design choices follow, each tied to a problem the broader agent and embodied-AI literature has already named.

[1] Standard RTS terms recur in the paper without redefinition: a *rush* commits early resources to a fast wave of cheap attackers; a *tech tree* is the dependency graph from cheap to advanced units; *kiting* is attack-while-retreating; an MCV (Mobile Construction Vehicle) deploys into a Construction Yard, the prerequisite for the production chain.

Three-state partial observability. The per-player fog of war distinguishes three cell states: *unexplored* (never seen), *fogged* (previously seen, currently out of sight, last contact cached), and *visible* (within the sight range of an agent unit). The agent operates over a world in which most cells are not directly observable on any given turn, and the decision to refresh contact information competes with every other action. This three-state primitive is the partial-observability assumption that the embodied-AI literature builds on (Yang et al., 2025; Huang et al., 2023), surfaced here as a first-class observation channel rather than a downstream consequence of sensor noise.

Real-time deliberation. Each decision turn advances roughly three seconds of game time. The world keeps moving while the agent reasons, harvesters continue along their routes, the opponent's rush continues to close, and a deferred decision is itself a decision. This is the property that distinguishes the setting from turn-based 4X benchmarks like CivRealm and CivBench (Qi et al., 2024; Chen et al., 2026), from Pokemon harnesses (Karten et al., 2026), and from Diplomacy harnesses descended from CICERO (Meta Fundamental AI Research Diplomacy Team (FAIR) et al., 2022; Guan et al., 2025), all of which pause the world while the agent thinks. The agent here must budget reasoning against opportunity cost.

Symmetric self-play with side-swap aggregation. Every adversarial match is played twice with seat assignments rotated, and the two halves are aggregated into a single match outcome. Without this, per-side win rates are confounded by spawn-side bias: Section 5.2 measures a 42 pp side gap on the canonical 1v1 map that side-swap aggregation neutralises. This extends pairwise evaluation methodology from open-ended chatbot ranking (Chiang et al., 2024) into a closed-loop adversarial setting where the asymmetry is structural rather than rater-side.

Machine-checked cheat resistance. Every scenario is gated by a deterministic validator: a no-op stall policy and a battery of wrong-strategy policies are run against every (pack, level, seed) tuple, and the scenario is admitted only if the intended capability policy wins and every shortcut policy loses. Roughly four hundred such tests run as a continuous-integration gate (Section 3.2). The benchmark therefore certifies, before any model

is run, that the headline metric cannot be hit by a shortcut policy that bypasses the targeted capability.

Beyond these four properties, the simulation core has no rendering, no audio, and no network dependency, which lets a single host run 64 or more concurrent episodes at sub-millisecond per-tick overhead. Determinism combined with this low-overhead parallelism is what makes the 15,500-episode sweep reproducible within a one-day wall-clock budget on a single host.

3.2 Scenario Authoring and the No-Cheat Bar

A scenario *pack* is a single YAML file that declares: a `meta` block (capability tag, real-world meaning, robotics analogue, benchmark anchors), a `base_map`, a `base` block (factions, starting cash, pre-placed actors, exposed tools), and a `levels` map with three required tiers (*easy*, *medium*, *hard*) plus deep-merge overrides. Each level supplies a `win_condition` and a `fail_condition`, both expressed in a declarative predicate grammar (composites `all_of`, `any_of`, `not`, `then` over leaf predicates including region containment, kill thresholds, deadlines, production targets, and discovery thresholds), plus a `max_turns` budget. Hard-tier seed rotation is implemented by per-actor `spawn_point` filtering.

Every pack must satisfy a binding *no-cheat / no-defect bar*:

> **Every lazy, brute, stall, shortest-path, or wrong-strategy policy must LOSE on every level and every hard seed (1–4). The intended capability policy must WIN. A non-win must be a real reachable timeout LOSS, never a draw.**

The negative half (every cheat must lose) is gated by a deterministic validator that runs an `observe`-only stall controller against every active pack $\times$ level $\times$ seed in $\{1, 2, 3, 4\}$. The positive half (intended capability must win) is gated by per-pack Python tests that pin an intended-policy win at one or two seeds and pin scripted cheat policies to losses at the same seeds. Roughly four hundred such tests run as a CI gate.

At v1.1 the suite contains 218 scenario packs grouped into eleven capability families (Figure 1). Three orthogonal `meta.capability` labels (*perception*, *reasoning*, *action*) plus an *adversarial* axis project the families onto the perception–reasoning–action chain.

Model	Provider	Episodes
Claude Opus 4.7	Bedrock	1,944
Claude Sonnet 4.6	Bedrock	1,944
Gemma 4 31B IT	OpenRouter	1,944
GPT-5.4	OpenAI	1,944
GPT-5.4-mini	OpenAI	1,944
Qwen 3.6-35B A3B	Together	1,944
Qwen 3.5-9B	Together	1,944
Kimi K2.5	Bedrock	1,944
Human testers	Play UI	77

Table 1: Model roster and episode counts.

4 Setup

Run methodology.

- *Standard scenario sweep.* Every $(pack, level)$ cell at a single seed, with three independent attempts per cell at τ=0.7 (pass^3). Default observation is the labelled minimap with fog of war. The bench exposes a 21-verb command interface (movement, attack, construction, special abilities, two no-op verbs).
- *Perception ablation.* Three observation channels (structured text only, text plus labelled minimap, image-primary) crossed with two fog states (fogged scoring cells, no-fog perception-control cells).
- *1v1 grid.* Five-model open-source cohort, all four hard-tier seeds per pair, both seat assignments aggregated under side-swap; raw per-side numbers are confounded (Section 5.2).
- *Handoff sweep.* Episodes prefixed with K scripted turns (stall, bad, good controller) to isolate recovery behaviour.
- *Human anchor.* 77 episodes by multiple testers across 52 hard-tier scenarios, played via the same Play UI, observation channel, and tool surface as the agents; an anchor, not a calibrated ceiling.

Metrics. Win rate, Composite, Tool Call Success Rate, Objective Progress, and Exploration Coverage are reported in Table 2. Each episode produces a composite score in $[0, 1]$ weighted over outcome, objective progress, exploration, discovery, combat, survival, tempo, and format/action

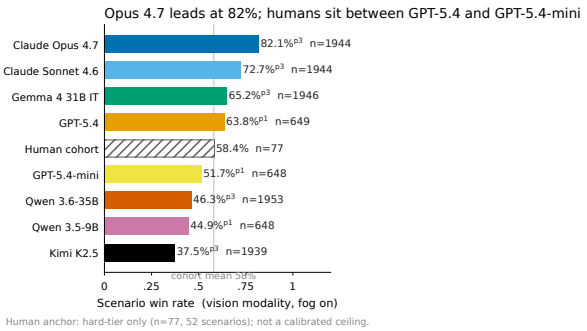

Figure 2: Aggregate win rate on the v1.1 scenario suite (vision modality, fog on). Bar annotations report *WR%, episode count* n, and *pass* mode (p3 = three reps per cell, p1 = one rep). The human cohort (hatched grey) plays a hard-tier subset and is an anchor, not a calibrated ceiling.

Model	WR	C	TCSR	Obj	Expl
Opus 4.7	82.1	73.9	99.8	86.7	29.6
Sonnet 4.6	72.7	68.0	100.0	79.3	32.7
Gemma 4 31B	65.2	62.8	99.6	73.2	27.0
GPT-5.4	63.8	62.0	99.8	71.2	33.7
Human anchor	58.4	—	—	69.3	40.4
GPT-5.4-mini	51.7	54.2	98.9	60.8	28.8
Qwen 3.6-35B	46.3	49.8	99.2	57.6	27.3
Qwen 3.5-9B	44.9	49.1	96.7	55.6	29.8
Kimi K2.5	37.5	43.0	99.7	48.1	25.1

Table 2: Overall performance on the v1.1 scenario suite. **TCSR**: fraction of commands accepted. **Obj**: worst predicate leaf's current/target at end of episode. **Expl**: map-revealed fraction. All per-episode means.

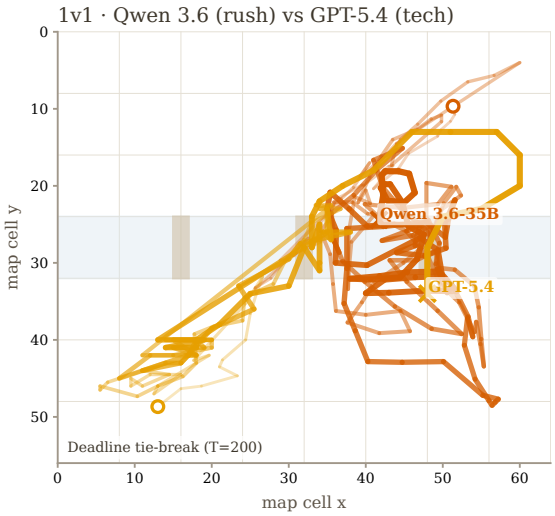

Figure 3: 1v1 trajectory overlay on the canonical adversarial map. Qwen 3.6-35B (committed rush) and GPT-5.4 (tech exploration) take qualitatively different paths from their respective spawns through the central bridge channel. Line opacity and width encode time (later turns are bolder); start positions are open circles, end positions bold $\times$ markers.

acceptance. Perception, reasoning, and action diagnostics are designer-chosen composites of the directly observable signals in Table 2: perception $= 0.7 \cdot$ exploration $+ 0.3 \cdot$ discovery; reasoning (on a non-win) $= 0.2 + 0.45 \cdot$ progress $+ 0.25 \cdot$ focus $+ 0.1 \cdot$ exploration; action $= 0.7 \cdot$ tool-call-acceptance $+ 0.3 \cdot$ non-idle-rate. Every predicate leaf in the win tree reports per-turn {target, current, ratio, satisfied}, and the *blocking ratio* is the worst leaf's value, which prevents a mean roll-up from hiding a hard-blocked objective. Pass^k reliability follows Yao et al. (2024); 1v1 cross-win-rate is fit with the Bradley-Terry pairwise model (Chiang et al., 2024).

5 Findings

5.1 Overall Performance

Figure 2 and Table 2 show the overall performance. Models generally distribute around the human baseline, with Opus 4.7 leading at 82.1% WR, Kimi K2.5 trails at 37.5%, a 44.6 pp range.

Tool-call success is saturated. In Table 2, TCSR is $\geq$ 99% for every model except Qwen 3.5-9B (96.7%). The 21-verb command vocabulary is mechanically accessible to all contemporary tool-use LLMs, and the action channel contributes no discrimination at the leaderboard scale.

Objective progress is the WR axis. Obj climbs monotonically with WR across the cohort, a 38.6 pp range (Opus 86.7% at the top, Kimi 48.1% at the bottom). It is the only Table 2 signal that sorts models in the same order as win rate.

Exploration coverage is narrow and non-monotonic. Expl spans only 8.6 pp across the cohort (Kimi 25.1% lowest, GPT-5.4 33.7% highest); GPT-5.4 explores most yet ranks fourth on WR. Many scenarios run on tick budgets short enough that full coverage is never required, so Expl saturates near a third of the map for everyone.

Humans sit in the middle of the model range. The human cohort wins 58.4% of its 77-episode hard-tier subset, between GPT-5.4 and GPT-5.4-mini. The frontier Claude models lead all human testers; the bottom four LLMs trail them. Human Obj (69.3) lands between the same two models, but human Expl (40.4) exceeds every model in the cohort, consistent with testers exploring more deliberately than current LLMs.

5.2 Where the Model Spawns Matters More Than Model Identity

On the canonical 1v1 map, which is rotationally symmetric, the corner the agent spawns at is a stronger predictor of outcome than which model is playing. A seat on the EAST half of the map wins 71% of seat-results; a seat on the WEST half wins 29%. The gap of 42 percentage points is larger than any pairwise gap between models in the same cohort.

The full breakdown of 390 seat-results from 195 matches across 15 model pairings is consistent across all four corners: the NE corner wins 61%, the SE corner 82%, the NW corner 19%, and the SW corner 39%. Within the same-model mirror pairs, where model identity is fully controlled, the same pattern appears in every pairing we ran: Qwen 3.6-35B wins 5/6 from the NE corner and 0/4 from the NW; GPT-5.4 wins 10/12 from the EAST half and 2/12 from the WEST. The gap holds at every difficulty tier (easy through hard) and in both the normal and side-swapped halves of the run protocol.

Implications for adversarial evaluation. Raw per-side 1v1 win rates are confounded by the spawn corner, and any 1v1 ladder built on a single side is uninterpretable. We report side-swap-aggregated outcomes only. The same confound doubles as a decisiveness signal: when a seed-pair flips outcomes across the two halves of the side-swap, the match is tight enough that the spawn corner decided it, and the aggregate win rate over four side-swapped seeds above 50% carries information about model identity.

5.3 Reproducible Means Hide Divergent Play

Claim. When the scenario, seed, and temperature are fixed, the same model can produce visibly different gameplay across repetitions. Yet aggregate win rate is stable to within 2.5 pp across pass counts, so the heterogeneity is invisible at the leaderboard layer.

The pass^3 sweep makes the gap explicit (Table 3). Win rate at pass@1 versus three-rep majority differs by at most 2.3 pp for every model in the cohort, and mean composite scores are similarly stable. The fraction of cells in which all three reps agree on the outcome, however, runs from 25.5% for Qwen 3.5-9B to 73.8% for Opus 4.7. Even the most consistent model disagrees with itself on roughly a quarter of cells at τ=0.7. The aggregate is reproducible; the per-cell behaviour is not.

Model	pass@1	3-rep maj.	Unanimous
Opus 4.7	82.3	82.1	73.8
Sonnet 4.6	71.9	72.7	60.3
Gemma 4 31B	64.1	66.2	51.0
Qwen 3.6-35B	46.9	46.3	32.9
Qwen 3.5-9B	42.6	44.9	25.5
Kimi K2.5	37.5	37.5	26.7

Table 3: Pass^3 stability. The gap between pass@1 and three-rep majority is $|\Delta| \leq 2.3$ pp for every model, yet unanimous-agreement rates differ by nearly 50 pp across the same cohort. All values are win-rate percentages.

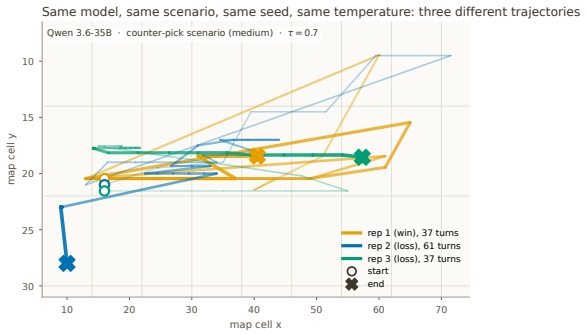

Figure 4: Pass^3 reps for Qwen 3.6-35B on a counter-pick scenario at the medium tier, seed 1, τ=0.7. Each line is the unit-centroid trajectory of one rep; the rep outcomes (1W/2L) appear in the legend. Line opacity and width encode time, with later turns drawn bolder. Open circles mark starting positions, bold X marks final positions.

Figure 4 shows what that within-cell variability looks like on the map for one Qwen 3.6-35B cell. Rep 1 commits early to a cautious base build-up, releases its scouts on a slow eastern sweep around turn 5, and reaches the kill quota by turn 36. Rep 2 races its scouts east on turn 1, runs them into the enemy garrison by turn 9, retreats the survivor to base coordinates near $(10, 23)$, and idles there for the next twenty-plus turns until the tick budget expires. Rep 3 splits the difference: an early eastern probe, an unforced retreat back to staging at $x \approx 15$, then a slow build of a five-unit force that pushes east again but never finds the eight kills. Same model, same prompt, same temperature, three different campaigns; statistically they average out to one number. This compression is what aggregate win rate hides, and recovering it is the central reason the bench logs per-turn state and reasoning. The losing reps here exhibit the same self-state perception gap we generalise to a family-level signature in Section 5.5.

5.4 Passivity Is Strategic, Not a Failure Mode

Most LLM benchmarks treat a "do nothing" turn as a failure signal; in RTS play it often is not. Stationing units at a defensible position or letting a harvester finish its route without issuing new orders is one of the most common winning behaviours in our data. Per-episode waiting rates across 7,843 episodes span a fivefold range, from 9.1% for Qwen 3.5-9B at the bottom to 50.0% **for the human cohort** at the top, with humans also outperforming five of the eight LLMs on the hard-tier subset. Within-model correlations between waiting rate and outcome are small and mixed in sign ($r \in [-0.13, +0.32]$), and every model becomes *more* active after a mid-game unit loss (action-vocabulary diversity rises from 2.52 to 3.24 distinct verbs); the data supports neither a freeze-under-adversity failure mode nor a simple "act more, win more" rule. What does separate winners from losers is whether each new turn issues something *new*: humans repeat the previous turn's command set 50.7% of the time, Qwen 3.5-9B only 22.6%, so low waiting is often bandwidth wasted on re-emitting orders to units already executing. We recommend reporting *command distinctness*, the fraction of action turns whose commands differ from the prior turn, alongside raw action volume.

5.5 Models Fail on the Easier Subtask

When a scenario demands both careful perception and careful reasoning, models systematically break on the easier of the two. The scenario's designer tag names the difficult step the scenario was built to probe; the rubric's loss attribution (Section 4) names the step where the model actually failed. They disagree symmetrically across the cohort, a finding about how models allocate capacity, not about the diagnostic axes. We walk through one losing episode of each kind (Figure 5).

Example A, a reasoning-heavy scenario. A long-horizon scenario asks the agent to stand up a refinery, unlock a war factory, field four armored units, then destroy the enemy headquarters under a single tick budget. The briefing supplies funds, base roster, and objectives in text; the difficulty lives in budgeting the cash chain and sequencing the tech prerequisites. We sample Qwen 3.6-35B (easy, seed 1, repetition 1; the rubric stamps the loss as a perception failure; objective progress 0.0). The model places the refinery on turn 3, then on turn 5 reasons "*the proc is still being built; I should wait*," and burns nineteen consecutive idle turns before the completed refinery becomes legible to it. Turn 23 finally re-reads the state correctly. The model spent its capacity on the planning chain; the easier subtask, reading whether the build had completed, dropped out, and the rubric correctly stamped perception.

Example B, a perception-heavy scenario. A scouting scenario hides the real enemy headquarters off-axis (north-west) and plants a decoy infantry picket on the obvious axis (east); the agent owns two reconnaissance units and wins by discovering the hidden base. The briefing explicitly warns that the eastern indicators are a feint. We sample the same model (medium, seed 1, repetition 1; rubric attribution reasoning; objective progress 0.0). The model issues seven turns of march commands that send both units east in lock-step toward the decoy, never splits the pair, and never sweeps the north-west corner. Every command is well-formed and target-valid; none advance the goal. The model read the briefing correctly; the easier subtask, acting on what it had read, dropped out, and the rubric correctly stamped reasoning.

These failures are not isolated. Across the eight-model cohort, on reasoning-tagged scenarios the weakest link is perception in 80–87% of strong-model losses (Opus, Sonnet, Gemma, Qwen 3.6, Kimi); on perception-tagged scenarios it is reasoning in 65–72% for four of five strong models (Gemma flips). The symmetric cross-tag pattern is the cohort-scale signature of capacity competition: the subtask the scenario invests its difficulty in absorbs the model's capacity, and the easier subtask is where the model breaks. The same shape recurs in the literature: LLMs articulate plans they cannot execute (Valmeekam et al., 2023b; Yao et al., 2023), self-reflection scaffolds exist because agents do not natively monitor outcomes (Shinn et al., 2023), and vision-language perception fails on inputs simple enough for downstream reasoning to remain intact (Tong et al., 2024).

Build-Tech is the cleanest instance. The same self-state perception gap that broke Example A scales up to a family-level signature. Every Build-Tech scenario requires the model to plan a multi-step prerequisite chain (reasoning) while tracking its own state (perception); the planning consumes the model's effective capacity and the easy self-state read drops out. The family is reasoning-tagged by design, yet across the five strong mod-

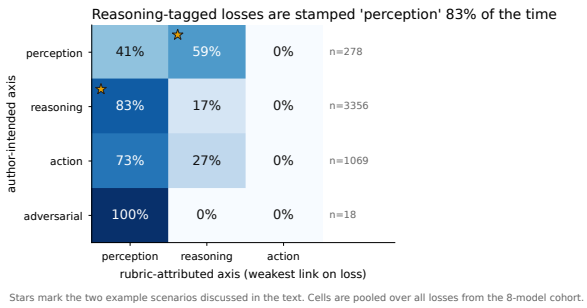

Figure 5: Author-intended axis (rows) versus rubric-attributed axis (columns) across all losses from the 8-model cohort. Off-diagonal mass dominates the reasoning and action rows. Stars mark the two example scenarios discussed in the text.

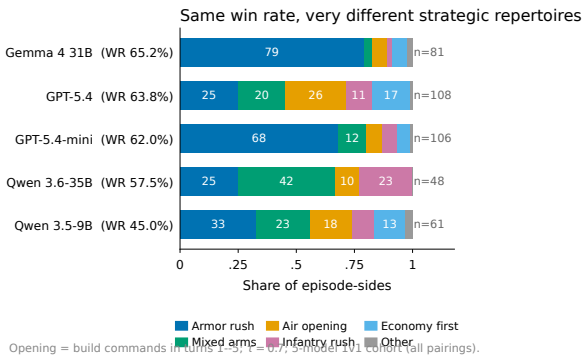

Figure 6: Per-model distribution of opening archetypes across all 1v1 pairings (five-model cohort, τ=0.7). Rows ordered by 1v1 win rate; segments labelled as percentages of all episodes. Gemma and GPT-5.4 are 1.4 pp apart on the leaderboard yet open very differently.

els it loses to a perception attribution 88–96% of the time (Opus 93%, Sonnet 84%, Gemma 89%, Qwen 3.6 93%, Kimi 96%): models re-emit a Build command across multiple turns because the tool return does not signal whether the previous build completed. Per-turn data surfaces what aggregate scores cannot.

5.6 Strategic Variance in 1v1 Play

Strategy variance is a model property, not a temperature effect. At fixed sampling temperature (τ=0.7) and starting state, some models commit to one opening in nearly every game while others draw fresh from a wide distribution. None of the cohort adapts its choice to opponent identity.

We classify each 1v1 episode by its first-five-turn build commands into one of six plain-English archetypes (Figure 6). Gemma 4 31B and GPT-5.4-mini commit to a single opening (79% and 68% Armor rush respectively); GPT-5.4 splits roughly evenly across five (Armor 25%, Air 26%, Mixed 20%, Economy 17%, Infantry 11%); Qwen 3.6-35B picks Mixed arms most often (42%); Qwen 3.5-9B is the most evenly spread. Gemma and GPT-5.4 sit at opposite ends of the variance axis yet their overall 1v1 win rates differ by only 1.4 pp (65.2% vs 63.8%): the same scalar outcome from clearly different policies. Strategic commitment also reorders the leaderboard: Qwen 3.6-35B ranks fourth of five on solo win-rate but first on 1v1 cross-win-rate (78%), because its rush commitment outperforms more exploratory openings in adversarial play (Spearman $\rho \approx +0.40$ between solo and 1v1 ranks). The two single-opening models keep their default against every opponent; the three exploratory models switch openings between opponents but the switches do not respond to opponent strength.

6 Conclusion

The five findings share a thread: aggregate win rate compresses a heterogeneous repertoire of behaviours into one scalar, and the per-turn artefacts make the repertoire legible. Two methodological implications follow. First, when a scenario demands both careful perception and careful reasoning, models break on the easier of the two (Section 5.5): the rubric's three-way attribution is doing its job, the model's capacity allocation is what fails. A diagnostic that names the easy-subtask drop directly may suit closed-loop agentic environments better than the canonical perception/reasoning/action triad. Second, passivity is not a unitary signal; we recommend *command distinctness*, the fraction of action turns whose commands differ from the prior turn, which separates purposeful waiting from re-emission redundancy.

We release the engine, the benchmark, the human anchor, and a live leaderboard. The deeper claim is methodological: in environments rich enough to admit many policies, the value of a benchmark is the repertoire it surfaces, not the leaderboard it produces. The benchmark's handoff sweep, predicate-grammar leaf telemetry, and 1v1 self-play substrate are designed to be training-data factories for the targeted SFT follow-ups we did not run here.

Limitations

Engine scope. The Rust engine is an RTS-Lite reimplementation: ground-only deep play, single-harvester-per-patch resource economy on the stan-

dard 1v1 map, and partial aerial / naval coverage. Documented out-of-scope features include helicopter passenger carry, naval ship-to-shore boarding, and dynamic ore contention. These limitations are explicitly enumerated in the engine documentation and the benchmark's scenario set does not test them.

Engine-side 1v1 slot bias. The two-player command channel exhibits a residual slot-2 economy bias: the player in slot 2 receives marginally more harvest throughput in the opening turns under otherwise-identical conditions. Section 5.2 measures this effect at the outcome level. The `-side-swap` protocol neutralises the bias at match aggregate (the aggregate outcome flips to draw when the two halves disagree), but per-side composite numbers on a single slot retain it; published 1v1 ladders should use side-swapped aggregates only.

One game, one map style. Red Alert Bench evaluates a single game with a single canonical 1v1 map and a single labelled-minimap render style. Generalisation across render styles is not measured. Our capability families are anchored to external benchmark families (PlanBench, ARC-Spatial, MicroRTS, ScreenSpot, ERQA) by design but are not validated by transfer experiments in this work.

Human baseline size and skill. Our human anchor (77 episodes, multiple testers) is small relative to the LLM evaluations and does not constitute a calibrated population ceiling.

Coverage asymmetry. Three of eight models (GPT-5.4, GPT-5.4-mini, Qwen 3.5-9B) ran pass^1 rather than pass^3 in the standard sweep; per-cell variance estimates for these models are not directly comparable.

Model release dates. The model versions evaluated here are those available at submission. Newer model releases may invalidate the specific rankings; the benchmark methodology is independent of any particular model snapshot.

References

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. 2025. τ^2-Bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*.

Kevin Black, Noah Brown, Danny Driess, and 1 others. 2024. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*.

John Chen and 1 others. 2026. CivBench: Progress-based evaluation for LLMs' strategic decision-making in Civilization V. *arXiv preprint arXiv:2604.07733*.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, and 1 others. 2024. Chatbot arena: An open platform for evaluating LLMs by human preference. In *International Conference on Machine Learning (ICML)*. ArXiv:2403.04132.

Anthony Costarelli, Mat Allen, Roman Hauksson, Grace Sodunke, Suhas Hariharan, Carlson Cheng, Wenjie Li, Joshua Clymer, and Arjun Yadav. 2024. GameBench: Evaluating strategic reasoning abilities of LLM agents. *arXiv preprint arXiv:2406.06613*.

Jie Gao, Kaiser Sun, Jen-tse Huang, Katherine Van Koevering, Sijie Ji, Heyuan Huang, Weiyan Shi, Zhuoran Lu, Ziang Xiao, Daniel Khashabi, and Mark Dredze. 2026. How to interpret agent behavior. *arXiv preprint arXiv:2605.13625*.

Alexander Duffy Guan and 1 others. 2025. Democratizing diplomacy: A harness for evaluating any large language model on full-press Diplomacy. *arXiv preprint arXiv:2508.07485*.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. 2023. VoxPoser: Composable 3D value maps for robotic manipulation with language models. In *Conference on Robot Learning (CoRL)*. ArXiv:2307.05973.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*. ArXiv:2310.06770.

Seth Karten and 1 others. 2026. The PokeAgent challenge: Competitive and long-context learning at scale. *arXiv preprint arXiv:2603.15563*. NeurIPS 2025 Competition Track.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, and 1 others. 2024. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*. ArXiv:2406.09246.

Zongyuan Li, Yanan Xu, Xiaojie Wen, Runnan Pan, and 1 others. 2024. LLM-PySC2: StarCraft II learning environment for large language models. *arXiv preprint arXiv:2411.05348*.

Dongmin Lim and 1 others. 2025. Orak: A foundational benchmark for training and evaluating LLM agents on diverse video games. *arXiv preprint arXiv:2506.03610*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, and 1 others. 2024. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations (ICLR)*. ArXiv:2308.03688.

Weiyu Ma, Qirui Mi, Yongcheng Yan, Yuqiao Wu, Runji Lin, Haifeng Zhang, and Jun Wang. 2024. Large language models play StarCraft II: Benchmarks and a chain of summarization approach. In *Advances in Neural Information Processing Systems (NeurIPS)*. ArXiv:2312.11865.

Julius Mayer, Mohamad Ballout, Serwan Jassim, Farbod Nosrat Nezami, and Elia Bruni. 2025. iVISPAR: An interactive visual-spatial reasoning benchmark for VLMs. In *EMNLP*. ArXiv:2502.03214.

Meta Fundamental AI Research Diplomacy Team (FAIR), Anton Bakhtin, Noam Brown, Emily Dinan, and 1 others. 2022. Human-level play in the game of Diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. GAIA: A benchmark for general AI assistants. *arXiv preprint arXiv:2311.12983*.

Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wołczyk, and 1 others. 2024. BALROG: Benchmarking agentic LLM and VLM reasoning on games. *arXiv preprint arXiv:2411.13543*.

Long Phan, Mantas Mazeika, and 1 others. 2025. TextQuests: How good are LLMs at text-based video games? *arXiv preprint arXiv:2507.23701*.

Jonah Philion, Amlan Kar, and Sanja Fidler. 2020. Learning to evaluate perception models using planner-centric metrics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Siyuan Qi, Shuo Chen, Yexin Li, Xiangyu Kong, Junqi Wang, and 1 others. 2024. CivRealm: A learning and reasoning odyssey in Civilization for decision-making agents. In *International Conference on Learning Representations (ICLR)*. ArXiv:2401.10568.

Xiao Shao, Weifu Jiang, Fei Zuo, and Mengqing Liu. 2024. SwarmBrain: Embodied agent for real-time strategy game StarCraft II via large language models. *arXiv preprint arXiv:2401.17749*.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*. ArXiv:2303.11366.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations (ICLR)*. ArXiv:2010.03768.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. 2024. Eyes wide shut? exploring the visual shortcomings of multimodal LLMs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. ArXiv:2401.06209.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023a. PlanBench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In *NeurIPS Datasets and Benchmarks Track*. ArXiv:2206.10498.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. 2023b. On the planning abilities of large language models: A critical investigation. In *Advances in Neural Information Processing Systems (NeurIPS)*. ArXiv:2305.15771.

Wichayaporn Wongkamjan, Feng Gu, Yanze Wang, Ulf Hermjakob, Jonathan May, Brandon M. Stewart, Jordan Boyd-Graber, and 1 others. 2024. More victories, less cooperation: Assessing Cicero's diplomacy play. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*. ArXiv:2406.04643.

Yue Wu, Xuan Tang, Tom Mitchell, and Yuanzhi Li. 2024. SmartPlay: A benchmark for LLMs as intelligent agents. In *International Conference on Learning Representations (ICLR)*. ArXiv:2310.01557.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, and 1 others. 2025. EmbodiedBench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*. ArXiv:2210.03629.

Alex L. Zhang and 1 others. 2025. VideoGameBench: Can vision-language models complete popular video games? *arXiv preprint arXiv:2505.18134*.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, and 1 others. 2024. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations (ICLR)*. ArXiv:2307.13854.

A Per-Family Performance Table

Table 4 reports per-family win rate for all eight evaluated LLMs across the ten non-adversarial capability families. Family ordering follows Figure 1 (Tactical Execution, Resource and Infrastructure, Perception and Scouting, Strategic Decision-Making). The bottom row is the model's overall WR across all 1,944 episodes.

The within-model coefficient of variation across families (Section 5.1) separates uniform-fingerprint models (Opus CoV 0.10, Gemma 0.12) from family-sensitive ones (Kimi 0.30, Qwen 3.5-9B 0.26); the rest cluster around 0.20.

B Pass^3 Reliability Detail

Table 5 expands the pass^3 summary in Section 5.3 with the per-pass aggregate win rate (pass@1 / pass^2 / pass^3) and the three per-cell reliability buckets (unanimous-win, at-least-one-win, zero-win).

C Human Annotator Details

We recruited 6 human annotators for this study. All annotators were fluent in English. The annotators' educational backgrounds ranged from Bachelor's degrees to PhDs, and the annotator pool included both cisgender female and transgender male participants. Each annotator received a $20 gift card as compensation for the annotation task.

The annotators were provided with the same instructions as the model and a graphical interface that could be interacted with via mouse clicks. All annotator's action on one game will be recorded into their trajectory.

D Pack-Level Discriminativeness Audit

A pack-level audit of all 648 scoring cells (216 packs $\times$ 3 difficulty levels) classifies each cell into four buckets, based on the spread of win rates across our eight models and the rank-correlation of those win rates with the aggregate ranking:

- *Saturated* ($99/648 = 15.3\%$): every model wins at least 90% of the time. These cells do not discriminate; they pass the no-cheat bar but contribute no signal to the leaderboard. Strong candidates for promotion to a "smoke-test" tier.

- *Floor* ($26/648 = 4.0\%$): every model wins at most 20% of the time. These cells test capability the cohort does not yet have. Examples: `build-sequence-tech-fastest`, `strict-sequence`, `econ-second-base-race:hard`.

- *Discriminative* ($263/648 = 40.6\%$): high spread across models AND Spearman $\rho \geq 0.5$ with aggregate ranking. The useful core of the benchmark.

- *Anti-correlated* ($30/648 = 4.6\%$): high spread but $\rho < 0$ with aggregate ranking; cells where weaker models occasionally beat the leaderboard top. Worth a separate review pass.

Per-family saturation rate (% of family cells where all eight models win $\geq 90\%$ of the time): Scouting & Vision 31.5%, Special Units 28.6%, Procedure 26.1%, Multi-front 18.8%, Tempo & Strategy 15.2%, Static Defense 13.6%, Combat Micro 13.1%, Long-Horizon 9.5%, Build & Tech 7.8%, Resource Economy 3.6%. Resource Economy is the family doing the most discriminative work; Scouting & Vision is the most saturated.

The fifteen hardest packs by cross-model mean WR (top to bottom): `adversarial-1v1-macro` (0.0%), `build-sequence-tech-fastest`, `strict-sequence`, `econ-protect-harvester-route`, `artofwar-indirect-approach`, `artofwar-lure-the-tiger`, `risk-blockade-bypass`, `scout-discover-hidden-base`, `combat-rocket-soldier-anti-vehicle`, `reasoning-frontier-commit`, `strategy-dilemma`, `econ-contested-expansion`, `strict-toolban-fidelity-under-pressure`, `tech-aggro-all-in`, `power-budget-online`. Of these fifteen, twelve are tagged `reasoning` in `meta.capability`: the field's collective wall is symbolic planning under spatial and resource constraints.

E Complete Scenario Inventory

Every active scenario pack in Red Alert Bench v1.1, grouped by capability family. Each pack ships at three difficulty levels (*easy*, *medium*, *hard*) and four hard-tier seeds. Pack names are stable identifiers used in the released artefacts at `openra_bench/scenarios/packs/<pack>.yaml`. Numbers in parentheses give the per-family count

Family	Opus 4.7	Sonnet 4.6	Gemma	GPT-5.4	GPT-5.4-m	Qwen 3.6	Qwen 3.5	Kimi K2.5
Combat Micro	75.0	70.6	71.0	62.4	53.6	42.9	47.6	32.9
Resource Economy	75.4	61.5	65.9	58.3	32.1	40.1	21.4	38.1
Static Defense	92.9	79.8	66.7	68.2	60.6	50.0	50.0	30.8
Scouting & Vision	90.1	88.3	74.3	68.4	61.4	69.5	70.2	57.3
Long-Horizon	96.0	82.5	53.2	66.7	47.6	41.3	47.6	29.4
Build & Tech	75.2	68.0	55.3	54.9	51.0	25.5	29.4	18.3
Procedure	85.0	73.4	62.8	66.7	56.5	50.2	49.3	45.4
Multi-front	85.0	77.8	74.4	65.2	49.3	49.3	47.8	48.3
Tempo & Adv.	75.0	58.7	55.7	53.6	47.6	42.4	44.0	33.3
Special Units	81.0	81.0	69.8	85.7	69.0	54.0	50.0	39.0
Overall	**82.1**	**72.7**	**65.2**	**63.8**	**51.7**	**46.3**	**44.9**	**37.5**

Table 4: Per-family win rate (%) by model on the ten non-adversarial capability families. The eleventh family (*adversarial*, one 1v1 pack) is reported separately in Section 5.6. Build & Tech is the universal wall for the bottom half of the leaderboard (Qwen 3.6-35B 25.5%, Qwen 3.5-9B 29.4%, Kimi 18.3%); Scouting & Vision is the great equaliser (Kimi reaches 57.3%, Qwen 3.5-9B reaches 70.2%). Per-family rank order does not always track overall rank: GPT-5.4 leads on Special Units (85.7%) despite ranking fourth overall, and the gap between Long-Horizon (96.0% Opus) and Build & Tech (75.2% Opus) is 21 pp within a single model.

Model	p@1	p^2	p^3	Unanimous	≥1	Zero
Opus 4.7	82.3	82.6	82.1	73.8	89.2	10.8
Sonnet 4.6	71.9	71.9	72.7	60.3	84.3	15.7
Gemma 4 31B	64.1	65.3	66.2	51.0	78.2	21.8
Qwen 3.6-35B	46.9	46.0	46.3	32.9	60.5	39.5
Qwen 3.5-9B	42.6	42.2	44.9	25.9	61.1	38.9
Kimi K2.5	37.5	37.8	37.5	26.7	49.8	50.2

Table 5: Pass^k reliability detail (%). Aggregate WR differs by less than 2.5 pp across passes for every model, but unanimous-agreement rates span 25.5% (Qwen 3.5-9B) to 73.8% (Opus): the gap between mean stability and per-cell variability is the behavioural-diversity signal of Section 5.3. Even Opus disagrees with itself on more than a quarter of cells at $\tau = 0.7$. "Unanimous" counts cells where all 3 reps win, "≥1" counts cells with at least one rep-win, "Zero" counts cells where 0/3 reps win.

(217 active packs total; one template file and three quality-archived packs are excluded).

Combat Micro (25 packs)

`combat-attack-from-behind-fog`, `combat-bait-counter-attack`, `combat-divide-and-conquer`, `combat-flanking-attack`, `combat-focus-fire-priority`, `combat-formation-tank-wedge`, `combat-harass-aggro-commit`, `combat-harass-balanced-hit-and-run`, `combat-heli-flank`, `combat-hold-chokepoint`, `combat-kite-and-pull`, `combat-kite-jeep-vs-tank`, `combat-naval-shore-strike`, `combat-pincer-coordination`, `combat-prevent-retreat`, `combat-protect-vip-escort`, `combat-retreat-after-engagement`, `combat-rocket-soldier-anti-vehicle`, `combat-skirmish-then-disengage`, `combat-stance-mgmt-attack`, `combat-suicide-charge-mission`, `combat-tank-vs-tank-engagement`, `combat-tanya-vs-rush`, `combat-target-priority-highvalue`, `combat-vehicle-vs-infantry-counter`.

Static Defense (19 packs)

`def-bridge-chokepoint`, `def-counter-battery`, `def-engineer-repair-under-fire`, `def-evacuation`, `def-in-depth`, `def-in-depth-vs-single`, `def-multi-direction`, `def-position-expected-direction`, `def-position-revealed-direction`, `def-pre-position-mobile-reserve`, `def-reinforce-the-breach`, `def-retreat-and-rebuild`, `def-stance-mgmt-hold-then-attack`, `def-surprise-flank-react`, `def-tower-line-vs-cluster`, `def-walls-vs-towers`, `def-while-building`, `def-with-ambush`, `defense-rush-survive`.

Multi-group Coordination **(9 packs)**

coord-converge-on-target, coord-cover-and-move, coord-diversionary-attack, coord-mutual-support, coord-relay-attack, coord-relay-vision-chain, coord-squad-handoff, coordination-ordered-rendezvous, coordination-staggered-window.

Resource Economy **(28 packs)**

econ-burn-rate-management, econ-buy-vs-build-decision, econ-cash-reserve-management, econ-contention-with-enemy, econ-contested-expansion, econ-deny-enemy-expansion, econ-expansion-timing, econ-far-patch-vs-near-patch, econ-harvester-defense-raid, econ-harvester-pathing-optimization, econ-mine-and-grow, econ-multi-patch-allocation, econ-overflow-to-silos, econ-protect-harvester-route, econ-quantitative-vs-qualitative-spend, econ-recover-from-zero-cash, econ-replace-dead-harvester, econ-resource-trade-with-self, econ-second-base-race, econ-silo-vs-spend, econ-startup-from-scratch, econ-target-cash-amount-by-deadline, econ-tech-vs-expand-decision, economy-force-buildup, economy-harvest-investment, economy-harvest-timebox, economy-investment, economy-time-box.

Build & Tech **(19 packs)**

build-defensive-skirt-corners, build-defensive-tower-cluster, build-defensive-tower-line, build-engineer-rebuild-after-loss, build-power-down-defensive, build-power-online-first, build-production-throughput-multibuilding, build-rally-point-management, build-repair-priority-under-fire, build-sell-and-rebuild-elsewhere, build-sequence-tech-cheapest, build-sequence-tech-fastest, build-sequence-tech-most-resilient, build-tech-skip-decision, building-and-planning, tech-aggro-all-in, tech-balanced-econ-then-tech, tech-production-planning, tech-turtle-defensive-tech.

Special Units **(12 packs)**

spec-engineer-capture, spec-nuke-strike, spec-spy-infiltrate, spec-tanya-c4-strike, spec-thief-steal-cash, tp-decision-under-clock, tp-pressure-procedural, tp-rush-multi-objective, tp-rush-objective-very-fast, tp-survive-and-grow, tp-survive-and-strike-at-window, tp-survive-n-turns.

Scouting & Vision **(18 packs)**

perception-count-the-threat, perception-count-the-threat-small-k, perception-frontier-reading, perception-target-vs-fog, scout-and-report, scout-and-survive, scout-count-defenders, scout-cycle-keep-info-fresh, scout-deny-enemy-vision, scout-detect-base-direction, scout-detect-enemy-tech, scout-detect-incoming-army, scout-discover-hidden-base, scout-far-frontier, scout-jeep-vs-infantry-cost-effective, scout-map-reveal-percent-target, scout-multiple-fog-areas, scout-track-enemy-movement.

Indirect Strategy **(10 packs)**

artofwar-indirect-approach, artofwar-lure-the-tiger, artofwar-sequenced-citadel, reasoning-frontier-commit, reasoning-risk-route, risk-blockade-bypass, strategy-dilemma, strategy-gauntlet, strategy-trilemma, strategy-twobody.

Long-Horizon Play **(21 packs)**

f11-defense-then-counter, f11-econ-tech-army-strike, f11-full-combined-arms, f11-pivot-on-scout, f11-rebuild-after-attrition, f11-vertical-strike-ground-air, f11-vertical-strike-naval, lh-100-turn-marathon-survival, lh-build-army-coordinate-multifront-attack, lh-credit-only-final-phase, lh-defense-tech-second-base, lh-econ-army-victory, lh-multi-checkpoint-5-plus, lh-opening-to-defense-to-counter, lh-opening-to-tech-to-army, lh-progression-stage-locked, lh-recovery-after-mid-game-loss, lh-scout-react-counter, lh-tech-pivot-attack, lh-tech-rush-vs-army-rush, longhorizon-opening-to-assault.

Tempo & Adversarial **(9 packs)**

adv-asymmetric-weaker-must-win, adv-rps-counter-pick, adversarial-1v1-macro, adversarial-duel, expansion-aggro-3-base-greedy, expansion-balanced-2-base-defended, expansion-turtle-1-base-fortified, tempo-double-window, tempo-strike-window.

Multi-front Resilience **(30 packs)**

harass-response-preserve, maint-repair-priority-order, maint-sell-and-recoup-cash, mcv-deploy-and-build, mcv-deploy-defensible-site, mcv-deploy-near-resource, mcv-deploy-relocate-under-pressure, mcv-deploy-second-base, mcv-deploy-third-base, mfb-base-1-defend-base-2-build, mfb-mirror-base-east-west, mfb-redundant-tech-buildings, mfb-rotating-production-pressure, mfb-supply-line-link-between-bases, mfb-tech-base-vs-economy-base, mfb-third-base-against-clock, mfb-two-base-simultaneous, mid-concede-vs-hold, mid-economy-under-fire, mid-tech-switch-on-scout, navigation-confined-hard-only, rob-cash-depletion-recovery, rob-deadline-shortened-midway, rob-multiple-simultaneous-pressures, rob-objective-change-midway, rob-objective-shift-with-or-clause, rob-partial-base-loss-continue, rob-unexpected-enemy-spawn, rob-unit-loss-recovery, rush-hour.

Procedure Robustness **(17 packs)**

action-multiunit-coordination, action-sequenced-execution, custom-map-no-enemy, power-budget-online, proc-checklist-no-deviation, proc-conditional-branch-action, proc-instruction-following-edge-case, proc-no-attack-passive-only, proc-only-build-no-combat, proc-only-defend-no-attack, proc-ordered-action-strict, proc-strict-toolban-fidelity, proc-tool-use-multi-distractor, proc-tool-use-with-distractor, strict-production-bom, strict-sequence, strict-toolban-fidelity-under-pressure.

F Within-Model 1v1 Opening Templates

Section 5.6 reports the within-model 1v1 opening-template analysis. Table 6 expands the per-model summary by listing the modal opening archetype each model produces against each of its five opponents, with the modal-vs-off-mode win-rate gap (positive: rational lock-in; negative: anti-rational lock-in).

G Engine Implementation Detail

The simulation core is a Rust reimplementation of the OpenRA *Red Alert* engine, distributed as a Python wheel built with `maturin`. Unit, weapon, and building data are byte-pinned to upstream OpenRA at commit `0938a27` (bleed branch); the YAML files are parsed at runtime via `include_str!`, so the wheel has no filesystem or network dependency on an external OpenRA checkout, and all actor and weapon statistics flow through a single entry point with a documented snapshot-bump procedure. The engine advances at approximately 90 simulation ticks per decision turn in

Model	Modal opening	Modal WR	Δ off-mode
Gemma 4 31B	Armor-Heavy	65%	+25
GPT-5.4-mini	Armor-Heavy	50%	+15
Qwen 3.6-35B	Mixed-Arms	67%	+1
GPT-5.4	Mixed-Light	32%	**−21**
Qwen 3.5-9B	Mixed-Arms	47%	+18

Table 6: Within-model 1v1 modal opening template (most-frequent archetype across the 8 seed-and-side-swap episodes per opponent, averaged over five opponents). Δ off-mode = modal WR minus off-mode WR. Gemma's ARMOR-HEAVY lock-in is rational (modal opening wins more than alternatives); GPT-5.4's MIXED-LIGHT lock-in is anti-rational (the model defaults to the opening with the worst win rate available to it).

non-interrupt mode (three internal frames per call, each frame roughly thirty ticks); the reachable maximum tick for a level with T turns is $93+90\cdot(T-1)$. The same interface that exposes one-turn-at-a-time deliberation for LLM agents also supports a gym-style RL interface; the benchmark uses the LLM-agent path, and the same engine has been used for population-based reinforcement-learning training in companion work.

YAML identifier glossary. Quoted transcript snippets in the body refer to units and buildings by their short YAML identifiers from the engine data files: `powr` (power plant), `proc` (ore refinery), `weap` (war factory, builds vehicles), `tent` (barracks, builds infantry), `hpad` (helipad), `dome` (radar dome), `harv` (harvester), `pbox` (pillbox, static defence), `e1` (rifle infantry), `e3` (rocket soldier), `1tnk`/`2tnk` (light/medium tank), `jeep` (fast scout vehicle), `hind` (helicopter gunship), `MCV` (Mobile Construction Vehicle, deploys into the Construction Yard).