AVA: Attentive VLM Agent for Mastering StarCraft II

Weiyu Ma[1,2*] **Yuqian Fu**[1*] **Zecheng Zhang**[3] **Bernard Ghanem**[4] **Guohao Li**[2†]
[1]Institute of Automation, Chinese Academy of Sciences [2]CAMEL-AI.org
[3]Strukto.ai [4]KAUST
sc2meisah@gmail.com, fuyuqian2022@ia.ac.cn, zecheng@strukto.ai,
bernard.ghanem@kaust.edu.sa, guohao.li@eigent.ai

Abstract

We introduce **AVACraft**, a multimodal StarCraft II benchmark supporting both Multi-Agent Reinforcement Learning (MARL) and Vision-Language Model (VLM) paradigms. Unlike SMAC-family environments that rely on abstract state representations and exclude VLMs, AVACraft provides RGB visuals, natural language observations, and structured state information, enabling systematic comparison between training-based and zero-shot methods across 21 scenarios spanning micromanagement, coordination, and strategic planning. We establish comprehensive baselines: six MARL algorithms (IQL, QMIX, QTRAN, VDN, MAPPO, IPPO) with Swin-Transformer backbones trained for 5M steps, and multiple VLMs including proprietary (GPT-4o) and open-source (Qwen3-VL) models. Results reveal complementary strengths—MARL peaks at 19.3% win rate after 5M steps, while VLMs achieve 75–90% zero-shot with human-aligned decisions—exposing trade-offs between training efficiency, performance ceilings, interpretability, and deployment cost. Code: https://github.com/camel-ai/VLM-Play-StarCraft2.

1 Introduction

Complex decision-making in dynamic, multi-agent environments represents a fundamental challenge in artificial intelligence, with StarCraft II emerging as a premier testbed due to its real-time nature, partial observability, and requirement for both tactical micromanagement and strategic coordination. Existing StarCraft II benchmarks, including SMAC (Samvelyan et al., 2019) and SMACv2 (Ellis et al., 2023), have facilitated significant advances in multi-agent reinforcement learning but suffer from two critical limitations. First, they rely on abstract feature representations that fundamentally diverge from human perception, creating an artificial gap between how AI agents and humans process battlefield information and limiting the ecological validity of learned behaviors. Second, these environments exclusively support traditional reinforcement learning approaches, lacking infrastructure for emerging Vision-Language Models (VLMs) that have demonstrated remarkable zero-shot reasoning capabilities across diverse domains. The rise of foundation models like GPT-4V and Qwen-VL has introduced a new paradigm for AI decision-making that operates without extensive task-specific training, yet their potential in complex, real-time strategic environments remains largely unexplored. While concurrent work such as LLM-PySC2 (Li et al., 2024) addresses macro-strategic decision-making and VS-Bench (Xu et al., 2025) evaluates strategic reasoning across multiple games, there remains an urgent need for benchmarks that systematically evaluate and compare both training-based MARL methods and zero-shot VLM approaches specifically for fine-grained tactical micromanagement on equal footing.

We introduce AVACraft, a multimodal StarCraft II benchmark that unifies two interaction paradigms in one framework. It supports RGB visuals, natural language, and structured state inputs, allowing both MARL algorithms and VLM-based agents to be evaluated under the same standardized setting. AVACraft includes 21 scenarios spanning basic control to complex multi-agent coordination, designed to expose the strengths and weaknesses of both approaches. We provide baselines for six MARL methods with Swin-Transformer backbones and multimodal fusion, and several VLMs, including GPT-4o, GPT-4-Turbo, Qwen3-VL-8B, and Qwen3-VL-30B, enabling fair comparison between trained MARL systems and zero-shot VLM agents.

Our experimental evaluation reveals complementary strengths between paradigms: MARL meth-

[*]Equal contribution. Work done during internship at CAMEL-AI.org.
[†]Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 4270–4290
July 2-7, 2026

ods achieve peak performance of 19.3% win rate through extensive training (up to 5M steps) even with state-of-the-art visual backbones, while VLMs demonstrate superior zero-shot capabilities with 75–90% win rates without any training, producing more interpretable and human-aligned decision processes as validated through expert evaluation involving professional StarCraft II players. The primary contributions of this work include:

- We design AVACraft, a multimodal benchmark environment for StarCraft II that supports both MARL and VLM decision-making paradigms through a unified observation space incorporating RGB visual inputs, natural language descriptions, and structured state information. Unlike existing benchmarks that focus on either macro-strategy (LLM-PySC2) or abstract multi-agent settings (VS-Bench), AVACraft targets fine-grained tactical micromanagement with full unit abilities.

- We establish comprehensive baseline implementations for both paradigms, including six MARL algorithms with Swin-Transformer backbones and multimodal input fusion (trained for 5M steps), and multiple VLMs covering both proprietary and open-source models, along with standardized evaluation protocols, cost analysis, and cross-modal ablation studies.

- We provide systematic empirical analysis across 21 carefully designed scenarios, revealing fundamental trade-offs between training-based optimization and zero-shot reasoning approaches, supported by expert human evaluation with statistical significance testing demonstrating superior human alignment of VLM-based decisions.

Beyond its immediate applications in gaming AI, AVACraft serves as a controlled testbed for studying the intersection of reinforcement learning and foundation models, opening new research directions for developing next-generation AI systems that combine the precision of MARL with the interpretability of VLMs.

2 AVACraft Benchmark Design

Traditional StarCraft II AI environments like SMAC and SMACv2, while advancing multi-agent reinforcement learning research, suffer from fundamental limitations that hinder the development of comprehensive AI evaluation frameworks. These environments employ abstract feature representations that create substantial perception gaps between AI agents and human players, often modifying unit attributes and employing "cheat mode" mechanics that deviate from authentic gameplay. Moreover, they exclusively support reinforcement learning paradigms, lacking the infrastructure necessary for evaluating emerging Vision-Language Models that demonstrate remarkable zero-shot reasoning capabilities. To address these limitations and establish a unified evaluation platform, we introduce AVACraft, a comprehensive multimodal benchmark that supports both traditional MARL approaches and modern VLM-based decision-making within a standardized framework.

Table 1: Comparison of StarCraft II environments. PySC2 provides the foundational API layer.

	SMAC	**SMACv2**	**PySC2**	**AVACraft**
Visual Input	×	×	Features	**RGB**
Language	×	×	×	✓
MARL Support	✓	✓	×	✓
VLM Support	×	×	×	✓
Enemy AI	Static	Procedural	Built-in	**Adaptive**
Abilities	Limited	Limited	Full	**Full**
Focus	Algorithms	Generalization	Full game	**Cross-paradigm**

× = not supported, ✓= supported, **Bold** = enhanced feature

AVACraft introduces three key design principles: **(1) Multi-modal observations** enabling fair comparison between MARL (RGB/scalar) and VLM (RGB+language) approaches; **(2) Complete unit abilities** preserving StarCraft II's tactical depth unlike SMAC's simplified mechanics; **(3) Adaptive opponents** preventing strategy exploitation through dynamic policy selection. Table 1 summarizes how AVACraft extends beyond existing environments to support cross-paradigm evaluation.

2.1 Environment Formulation

We formalize AVACraft as a Partially Observable Markov Decision Process (POMDP) defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, P, R, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{O}$ is the observation space, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1]$ is the discount factor. At each timestep t, agents receive partial observations $o_t \in \mathcal{O}$ derived from the true state $s_t \in \mathcal{S}$ and select actions $a_t \in \mathcal{A}$. Note that AVACraft strictly maintains partial observability: the RGB screen and minimap

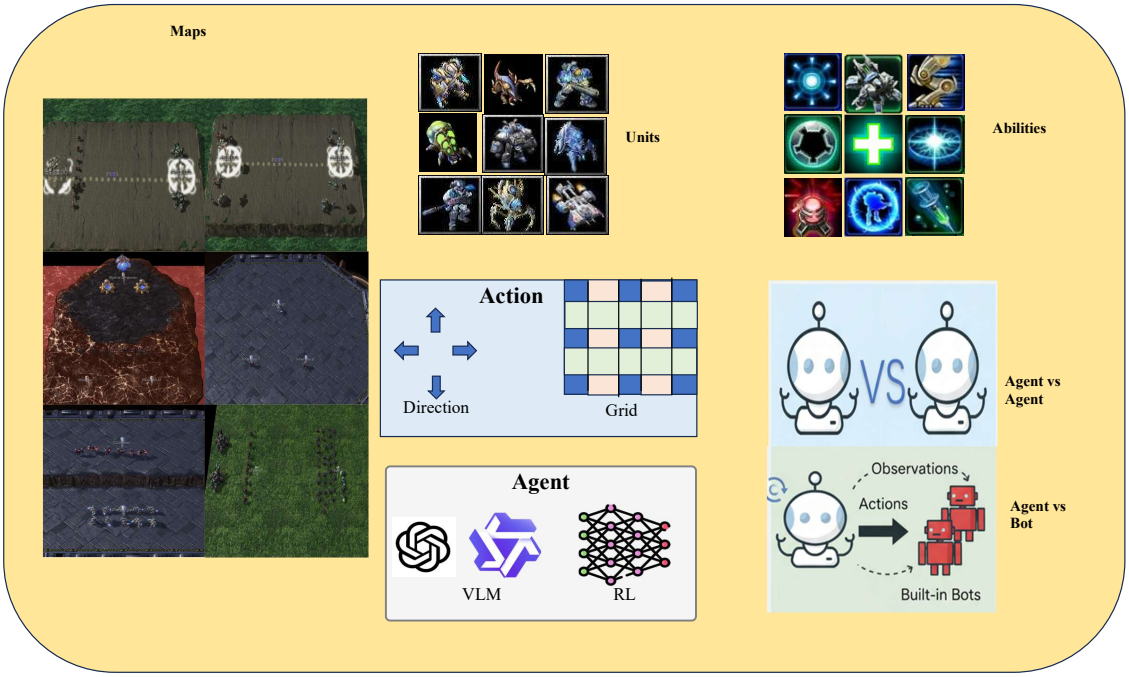


Figure 1: **AVACraft** environment.

observations natively obey StarCraft II's Fog of War mechanics, so agents only receive visual and textual information within allied units' sight range. The key innovation of AVACraft is providing *flexible observation modes* within $\mathcal{O}$ to support both MARL and VLM paradigms while maintaining a unified evaluation framework.

2.2 Unified Observation Framework

The observation space $\mathcal{O}$ provides flexible representations tailored to different AI paradigms while maintaining consistency across evaluation scenarios. We define four observation modes summarized in Table 2.

Table 2: Observation modes in AVACraft

Mode	Notation	Primary Use Case
RGB Visual	o_t^{rgb}	CNN/Transformer-based MARL
Scalar Features	o_t^{feat}	SMAC-compatible MARL
Hybrid	o_t^{hybrid}	Multimodal MARL research
VLM-Optimized	o_t^{vlm}	Vision-Language Models

RGB Visual Mode provides human-like visual observations:

$$o_t^{\text{rgb}} = (I_t^{\text{scr}}, I_t^{\text{mini}}) \tag{1}$$

where $I_t^{\text{scr}} \in \mathbb{R}^{H_s \times W_s \times 3}$ captures the main battlefield view (default: $H_s = 160, W_s = 120$) and $I_t^{\text{mini}} \in \mathbb{R}^{H_m \times W_m \times 3}$ provides tactical overview (default: $H_m = W_m = 32$). Resolutions are configurable to balance visual fidelity and computational efficiency.

Scalar Feature Mode maintains compatibility with existing MARL research through vector representation $o_t^{\text{feat}} \in \mathbb{R}^d$ containing unit attributes (health, shields, position, cooldowns) in SMAC-compatible format, enabling direct comparison with prior work.

Hybrid Mode combines visual and structured information:

$$o_t^{\text{hybrid}} = (I_t^{\text{scr}}, I_t^{\text{mini}}, o_t^{\text{feat}}) \tag{2}$$

enabling research into multimodal MARL approaches that leverage both spatial reasoning from images and explicit feature information.

VLM-Optimized Mode augments visual input with linguistic context:

$$o_t^{\text{vlm}} = (I_t, T_t, \mathcal{U}_t) \tag{3}$$

where:

- I_t: high-resolution RGB screenshot for visual grounding
- T_t: natural language description of battlefield state, tactical situation, and mission objectives
- $\mathcal{U}_t = \{u_1, \ldots, u_n\}$: structured metadata for each unit where $u_i = (\text{id}_i, \text{type}_i, \text{pos}_i, \text{hp}_i, \text{status}_i)$

This multimodal representation lets VLMs use their pre-trained visual reasoning without task-specific training. Natural language descriptions are generated at each timestep to provide precise numerical details, such as HP, shields, and cooldowns, that may be unclear from RGB inputs alone.

2.3 Action Space Design

The action space $\mathcal{A}$ supports fine-grained tactical control through three complementary categories:

$$\mathcal{A} = \mathcal{A}_{\text{atk}} \cup \mathcal{A}_{\text{mov}} \cup \mathcal{A}_{\text{abl}} \tag{4}$$

Attack Actions ($\mathcal{A}_{\text{atk}}$): Ordered pairs (i, j) specifying unit i targeting enemy unit j, enabling focus-fire and target prioritization strategies critical for tactical micromanagement.

Movement Actions ($\mathcal{A}_{\text{mov}}$): Support both precise and directional positioning:

- *Grid positioning*: (i, x, y) where $(x, y) \in [1, 10]^2$ provides discrete spatial coordinates for battlefield coordination
- *Directional movement*: (i, d) where $d \in \{\text{UP, DOWN, LEFT, RIGHT}\}$ enables rapid repositioning like SMAC

Ability Actions ($\mathcal{A}_{\text{abl}}$): Triples $(i, \text{ability}, \text{target})$ covering key StarCraft II tactics, including defensive, offensive, and mobility abilities. Unlike SMAC, AVACraft retains unit abilities, preserving the tactical richness of high-level gameplay. Targets may be positions, unit IDs, or null, depending on the ability.

2.4 Adaptive Enemy Policies and Standardized Evaluation

To ensure robust evaluation across different challenge levels, AVACraft implements an adaptive enemy system extending beyond traditional static AI opponents. Drawing inspiration from SMAC-Hard (Deng et al., 2024b), we develop a multi-tier enemy policy framework:

Built-in AI: StarCraft II difficulty level 7 (Very-Hard) provides a consistent and competent baseline opponent.

Script-based Policies: Three specialized behavior policies per scenario generated through LLM-assisted behavior tree synthesis, each emphasizing different tactical approaches (aggressive rushing, defensive positioning, economic optimization). This diversity prevents agents from overfitting to single strategies.

Randomized Selection: Dynamic policy selection during evaluation ensures generalization assessment by preventing exploitation of predictable opponent patterns.

Our benchmark encompasses 21 carefully designed scenarios spanning multiple complexity dimensions: 12 core micromanagement scenarios testing fundamental tactical skills (unit control, target prioritization, ability timing), 5 coordination scenarios requiring multi-unit synchronization and formation control, and 4 strategic scenarios incorporating terrain utilization and resource management. Each scenario supports both PvE and PvP modes, with PvE scenarios featuring the adaptive enemy system and PvP scenarios enabling direct agent-versus-agent competition for comparative evaluation between paradigms. Detailed scenario specifications are provided in Appendix F.

AVACraft employs a sparse reward structure $R(s_t) \in \{-1, 0, 1\}$ corresponding to defeat, ongoing/draw, and victory states respectively. Episodes terminate under three conditions: complete enemy elimination (victory), complete allied elimination (defeat), or 300-second time limit (draw). This design provides clear performance signals while maintaining tactical flexibility and avoiding reward shaping that might bias particular approaches.

3 Baseline Implementations

To establish comprehensive baselines for both paradigms supported by AVACraft, we implement representative approaches for Vision-Language Model agents and Multi-Agent Reinforcement Learning algorithms. These baselines serve as reference implementations for the research community and demonstrate the benchmark's capability to fairly evaluate fundamentally different decision-making approaches.

3.1 VLM-based Decision Making: Attentive VLM Agent (AVA)

We develop AVA as our primary VLM baseline, designed to leverage the multimodal reasoning capabilities of foundation models for strategic decision-making. The AVA architecture integrates three key components: a Multimodal Priority Inference mechanism for strategic unit targeting, a knowledge-enhanced decision system through retrieval-augmented generation, and a dynamic role assignment framework for coordinated multi-agent behavior. We emphasize that AVA is a

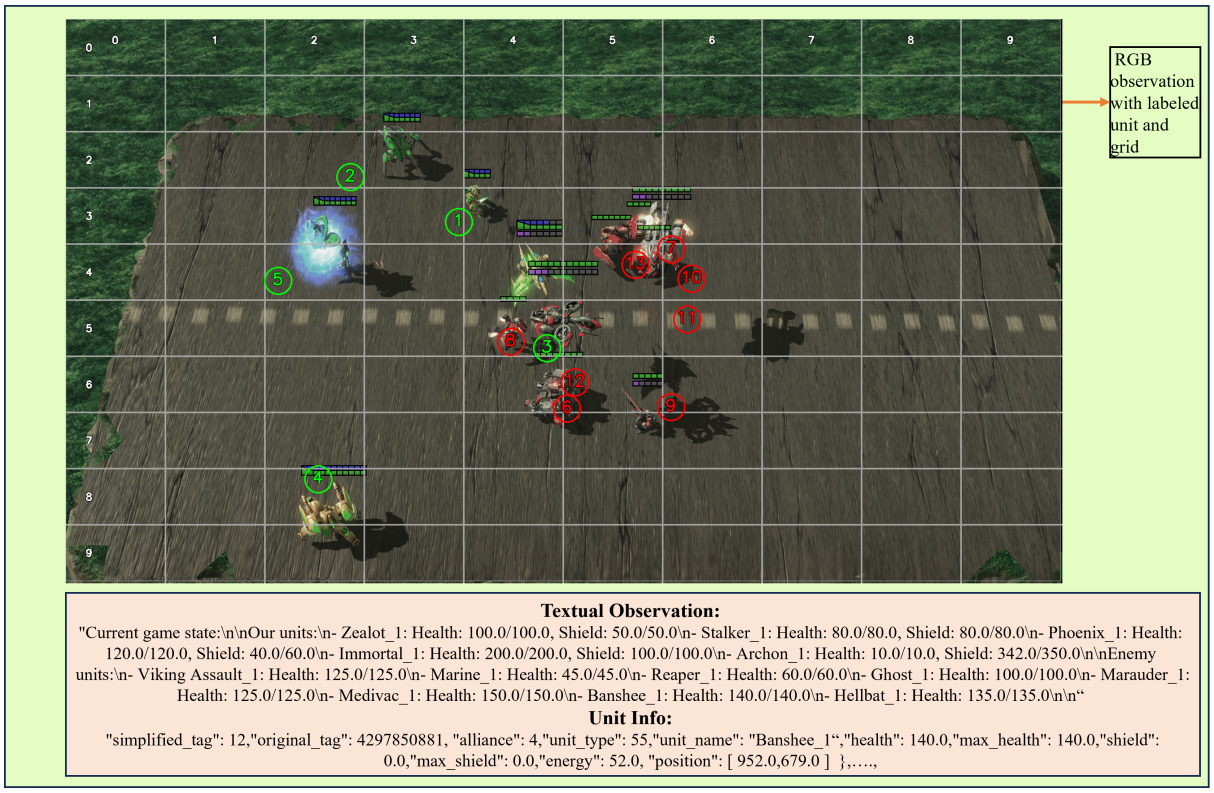


Figure 2: Observation Space of **AVACraft** environment.

proof-of-concept baseline that validates the environment's capabilities for evaluating VLM-based agents, rather than a novel architectural contribution.

3.1.1 Multimodal Priority Inference Mechanism

Our priority inference system processes battlefield information through structured skill planning and tactical decision-making. The mechanism operates in two key stages to identify and prioritize strategic elements. First, we implement a VLM Planner that evaluates the battlefield situation and generates specific micro-management skill plans:

$$S = \text{VLM}_{\text{plan}}(I, T, H) = \{s_{\text{primary}}, s_{\text{secondary}}\}, \quad (5)$$

where the planner outputs structured skill plans with primary and secondary tactical objectives. Based on the planner's output, the system performs precise unit identification and classification:

$$A = \text{VLM}_{\text{detect}}(I) = \{a_1, ..., a_n\}, \quad (6)$$

where each annotation $a_i = (p_i, c_i, b_i)$ consists of unit position p_i, unit class c_i, and bounding box b_i for accurate spatial localization.

The critical Multimodal Priority Inference process then integrates visual features with tactical objectives through skill-aware natural language prompting:

$$U_{\text{priority}} = \text{VLM}_{\text{analyze}}(I, T, H, A, Q, S), \quad (7)$$

where I is the current game screenshot, T is the text state description, H represents action history for temporal reasoning, A is the set of unit annotations, Q is the tactical analysis prompt generated based on the skill plan, and S is the current skill plan from the VLM Planner. The VLM outputs its analysis in structured natural language, integrating battlefield assessment with tactical prioritization.

3.1.2 Knowledge Integration through RAG

To enhance tactical decision-making with domain expertise, we implement a Retrieval-Augmented Generation system that operates on priority units identified through Multimodal Priority Inference. Given the priority unit set $U_{\text{priority}} \subseteq A$, we formulate the knowledge retrieval and integration process as:

$$K(u) = \text{Retrieve}(c_u) = \{s_u, m_u, t_u\} \quad \forall u \in U_{\text{priority}}, \quad (8)$$

where for each unit u with class c_u, we retrieve a knowledge tuple $K(u)$ consisting of unit specifications s_u, matchup data m_u, and tactical insights t_u.

The retrieved knowledge is then integrated with the current game state through a context-aware generation process:

$$D = \text{VLM}_{\text{synthesize}}(I, T, H, U_{\text{priority}}, \{K(u)\}), \tag{9}$$

where D represents the tactical decision guidance generated by combining retrieved knowledge with current game state representation.

3.1.3 Dynamic Role Assignment and Decision Pipeline

For multi-agent coordination, we implement a dynamic role assignment framework that adapts to evolving battlefield conditions. Let $\mathcal{N} = \{1, ..., N\}$ denote the set of agents and $\mathcal{Z} = \{z_1, ..., z_m\}$ represent possible roles. The role assignment function $\phi : \mathcal{N} \to \mathcal{Z}$ maps each agent to a specific role, optimized through a utility function $U(\phi, s)$ that evaluates role effectiveness given the current state $s \in \mathcal{S}$. Our framework leverages VLMs through a multimodal fusion function $z_i = \text{VLM}_{\text{role}}(I, T, C)$, where the model processes visual inputs I, textual prompts T, and contextual information C to generate role assignments.

The complete decision-making pipeline maps POMDP observations to actions through VLM-based transformations. At each timestep t, given observation $o_t = (I_t, T_t, U_t)$ from AVACraft environment state $s_t \in \mathcal{S}$, our system generates actions by maintaining a history buffer H_t and processing each step to maximize the trade-off between strategic depth and real-time responsiveness.

3.2 MARL-based Decision Making

For the MARL paradigm, we implement six representative algorithms spanning both value-decomposition and policy-gradient methods: Independent Q-Learning (IQL), QMIX, QTRAN, Value Decomposition Networks (VDN), Multi-Agent PPO (MAPPO) (Yu et al., 2022), and Independent PPO (IPPO). These algorithms are adapted to work with AVACraft's visual observations through state-of-the-art visual processing architectures.

Our upgraded MARL implementation employs a **Swin-Transformer** (Swin-Tiny, 27.5M parameters) as the visual backbone, replacing the simple CNN used in preliminary experiments to ensure that RL agents have access to competitive visual feature extraction. The visual encoder processes both screen (160×120) and minimap (32×32) observations through separate Swin-Tiny streams with adaptive pooling to produce fixed-size representations.

We also introduce a **multimodal fusion** mode for MARL, combining visual features with pre-computed text embeddings. Natural language observations are encoded by GTE-Base and fused with visual features through a learned projection layer before entering the mixing networks, enabling direct comparison with VLMs using the same textual information.

The algorithms differ in their coordination mechanisms: IQL treats agents independently, QMIX uses a monotonic mixing network, QTRAN relaxes the monotonicity constraint, VDN sums individual Q-values, while MAPPO and IPPO use centralized and independent critic architectures respectively with PPO policy updates. All implementations support the observation modes provided by AVACraft (RGB visual, SMAC-compatible scalar, hybrid, and vision+text), enabling comparative analysis of how different input modalities affect learning efficiency and final performance.

4 Experimental Evaluation

We conduct comprehensive experiments to evaluate both VLM and MARL paradigms within the AVACraft benchmark, focusing on four primary objectives: (1) establishing performance baselines for both paradigms across diverse scenarios, (2) conducting systematic cross-paradigm comparison with cross-modal ablations, (3) validating human alignment through expert evaluation with statistical testing, and (4) analyzing computational costs and scalability. Our experimental setup leverages dual A100 40GB GPUs for MARL training and the Camel framework for VLM agent coordination, with all experiments conducted at 2Hz frequency to balance strategic decision depth and computational efficiency. Detailed hyperparameters are provided in Appendix D.

4.1 Cross-Paradigm Performance Analysis

We evaluate both paradigms across a representative subset of AVACraft scenarios. For MARL evaluation, we implement six algorithms using RGB visual observations processed through Swin-Tiny backbones and train for **5 million steps** on the foundational 3m scenario, following standard SMAC practice. We evaluate MARL agents under both Vision-Only and Vision+Text input modes. For VLM evaluation, we assess multiple models includ-

Table 3: Cross-paradigm performance comparison on the 3m scenario. MARL results after 5M training steps with Swin-Tiny backbone. Win rates reported as mean $\pm$ std over 5 seeds. †Vision+Text mode uses GTE-Base text embeddings fused with visual features.

Method	Input Mode	Steps	Win Rate (%)
MARL Methods (Swin-Tiny backbone)			
MAPPO	Vision+Text†	5M	19.3 ± 3.2
IPPO	Vision Only	5M	18.2 ± 2.8
IPPO	Vision+Text†	5M	16.6 ± 3.5
QMIX	Vision Only	5M	27.1 ± 4.1
QTRAN	Vision Only	5M	2.0 ± 1.4
IQL	Vision Only	5M	0.0 ± 0.0
VDN	Vision Only	5M	0.0 ± 0.0
VLM Methods (Zero-shot, proprietary)			
GPT-4o	VLM-Optimized	0	**81** ± 3.9
GPT-4-Turbo	VLM-Optimized	0	79 ± 4.1
GPT-4o-mini	VLM-Optimized	0	76 ± 4.3
Qwen-VL-Plus	VLM-Optimized	0	75 ± 4.3
VLM Methods (Zero-shot, open-source)			
Qwen3-VL-30B	VLM-Optimized	0	50 ± 5.0
Qwen3-VL-8B	VLM-Optimized	0	40 ± 4.9

ing proprietary (GPT-4-Turbo, GPT-4o, GPT-4o-mini) and open-source (Qwen-VL-Plus, Qwen3-VL-8B, Qwen3-VL-30B) models across 12 micromanagement scenarios.

Table 3 reveals striking differences between paradigms on the foundational 3m scenario. Even with upgraded Swin-Transformer backbones and 5M training steps, MARL methods achieve at most 27.1% win rate (QMIX), with policy-gradient methods (MAPPO, IPPO) plateauing around 16–19%. In contrast, VLM approaches demonstrate superior zero-shot capabilities, with proprietary models achieving 75–81% win rates and open-source Qwen3-VL models achieving 40–50% win rates without any training, highlighting the power of pretrained foundation models for strategic reasoning.

Cross-Modal Ablation. An important finding emerges from the MARL cross-modal ablation: IPPO with Vision+Text input (16.6%) slightly underperforms IPPO with Vision-Only input (18.2%), suggesting that from-scratch MARL agents struggle to effectively fuse pre-computed text embeddings with visual features during training. In contrast, VLMs inherently align text and image modalities through pre-training, naturally leveraging the natural language channel that provides exact numerical values (HP, cooldowns, unit IDs) critical for precise tactical decisions. This ablation demonstrates AVACraft's utility in exposing the unique cross-modal alignment capabilities of foundation models versus traditional RL architectures.

Table 4: VLM performance across AVACraft scenarios. Win rates (%) for zero-shot evaluation (mean $\pm$ std over 20 episodes).

Scenario	GPT-4o	Qwen-VL	Q3-30B	Q3-8B	Challenge
Low Complexity					
3m	**81**	75	50	40	Coordination
2m_vs_1z	**23**	10	15	5	Micro control
Medium Complexity					
mixed_units	**79**	75	60	35	Targeting
2s3z	**41**	25	20	10	Unit synergy
3s_vs_3z	**32**	10	15	5	Positioning
2s_vs_1sc	**5**	0	0	0	Range mgmt.
High Complexity					
pvz_ht	**34**	25	20	10	Ability timing
8m2st_vs_35zg4b	**53**	25	30	15	Formation
8m1mv_vs_2st	**12**	0	5	0	Support coord.
8m_vs_2pc1wp	**11**	0	5	0	Terrain
6r_vs_8z	0	0	0	0	Hit-and-run
Very High Complexity					
2c_vs_64zg	0	0	0	0	AOE optim.
Average	**30.9**	20.4	18.3	10.0	

Table 4 shows that both proprietary and open-source VLMs perform well on low- to medium-complexity scenarios, but struggle on harder ones. GPT-4o and Qwen-VL achieve strong results on tasks like mixed_units and 3m, while Qwen3-VL-30B is competitive on simpler settings, supporting reproducible evaluation without proprietary APIs. However, all models fail on the most difficult scenarios, with 0% win rates on 2c_vs_64zg and 6r_vs_8z.

Error Analysis of 0% Win Rate Scenarios. The universal 0% win rate on 2c_vs_64zg and 6r_vs_8z across all VLMs—including Qwen3-VL-30B which achieves 90% on simpler maps—confirms that these failures reflect the true capability ceiling of current VLMs for dense spatial reasoning and high-frequency micro-control, rather than environment bugs or prompt engineering issues. Specifically, 2c_vs_64zg requires precise AOE line-damage optimization against 64 units with continuous kiting, while 6r_vs_8z demands sustained hit-and-run micro over many timesteps. Both require spatial precision and temporal consistency that exceed current VLM capabilities.

4.2 Architectural Component Analysis

To understand the contribution of different components in our AVA baseline, we conduct a comprehensive ablation study using GPT-4-Turbo on the mixed_units scenario. The three key components are: (1) Dynamic Role Assignment for coordination, (2) Multimodal Priority Inference (MPI) for target selection, and (3) Retrieval-Augmented Generation (RAG) for domain knowledge integration.

Table 5: Component ablation study showing individual and combined contributions of AVA architecture components.

Role	MPI	RAG	Win Rate (%)
✓	✓	✓	**87** $\pm$ 3.4
✓	✓	-	71 $\pm$ 4.5
✓	-	✓	65 $\pm$ 4.8
-	✓	✓	70 $\pm$ 4.6
✓	-	-	24 $\pm$ 4.3
-	✓	-	50 $\pm$ 5.0
-	-	✓	26 $\pm$ 4.4
-	-	-	20 $\pm$ 4.0

Table 6: Head-to-head VLM comparison on mixed_units scenario (20 matches per pairing).

Model	GPT-4o	GPT-4T	Qwen	Gemini-F	Win Rate
GPT-4o	–	9:11	13:7	9:11	55%
GPT-4-Turbo	11:9	–	14:6	8:12	58%
Qwen-VL	7:13	6:14	–	8:12	35%
Gemini-Flash	11:9	12:8	12:8	–	62%

The ablation results (Table 5) demonstrate the complementary nature of AVA's components. The complete system achieves 87% win rate, with MPI providing the most substantial individual contribution (50% vs 20% baseline), RAG contributing 20–25% improvement through domain knowledge, and Role Assignment adding 15–20% through enhanced coordination. Notably, all components show positive interactions, with the combined system significantly outperforming any individual component.

4.3 Human Alignment Evaluation

To assess the human-like qualities of decision-making across paradigms, we conduct a structured evaluation with seven participants representing diverse StarCraft II expertise levels: one professional player, two Master-level players, one Diamond-level player, one Platinum-level player, one Gold-level player, one novice, and one spectator. Participants evaluate both VLM (AVA) and MARL agents across three metrics on a 1–5 scale: Game Bug Exploitation (higher indicating less exploitation), Reasoning Coherence, and Human Similarity. Evaluations were conducted in a blinded setting where participants were not informed which agent type they were evaluating. See Appendix G for detailed metric definitions.

Table 7 shows VLM agents significantly outperforming MARL approaches across all metrics and expertise levels ($p < 0.01$, Mann-Whitney U test). Expert evaluators were particularly critical of MARL agents, noting frequent exploitation of environment mechanics and poor strategic coherence (average scores of 1.3–2.0). In contrast, VLM agents received consistently high ratings (4.3–4.5 average) for producing interpretable, human-like decision processes. Expert evaluators specifically highlighted VLM agents' implementation of advanced tactical principles including armor-type targeting, focus-fire coordination, and formation control that closely resemble professional gameplay strategies.

Table 7: Human evaluation comparing VLM and MARL approaches. Scores: 1–5 scale (higher is better). Statistical significance tested via Mann-Whitney U test.

Evaluator Group	Metric	MARL	VLM
Expert (n=3)	Bug Exploit.*	1.3	**5.0**
	Reasoning	2.0	**4.3**
	Human Sim.	1.0	**4.7**
Mid-tier (n=3)	Bug Exploit.*	2.0	**3.7**
	Reasoning	2.0	**4.7**
	Human Sim.	1.3	**4.7**
Novice (n=2)	Bug Exploit.*	2.5	**4.0**
	Reasoning	2.5	**3.5**
	Human Sim.	3.0	**4.0**
Overall Average		1.9	**4.4**††

*Higher = less bug exploitation. ††Mann-Whitney U test: $p < 0.01$ for all three metrics (VLM vs MARL overall).

4.4 Computational Cost and Scalability Analysis

We analyze the computational requirements and deployment costs of both paradigms to provide practical guidance for researchers and practitioners.

Table 8 reveals fundamental cost trade-offs between paradigms. MARL training requires approximately 65–80 hours on dual A100 GPUs for 5M steps on the 3m scenario, with significant memory requirements for experience replay buffers. Once trained, MARL agents achieve fast inference ($\sim$8–10ms per step). VLM agents operate with zero training overhead but incur per-decision costs: GPT-4o averages 2.3 seconds and $\sim$\$0.027 per decision, resulting in $\sim$\$4.05 per episode. Open-source Qwen3-VL models eliminate API costs entirely and can be deployed on a single A100 GPU, making fully reproducible evaluation accessible without proprietary dependencies. For scenarios requiring rapid deployment or frequent scenario

Table 8: Computational cost comparison between paradigms. MARL costs reflect 5M-step training on dual A100 40GB GPUs. VLM costs reflect per-episode inference on the 3m scenario (avg. 150 steps/episode).

Method	Training Time	Latency (s/step)	Tokens/ Decision	Cost/ Episode
MARL Methods				
MAPPO (Swin-T)	∼72h	0.008	–	GPU only
IPPO (Swin-T)	∼65h	0.008	–	GPU only
QMIX (Swin-T)	∼80h	0.010	–	GPU only
VLM Methods (Proprietary)				
GPT-4o	0	2.3	∼1,800	∼$4.05
GPT-4-Turbo	0	3.1	∼2,100	∼$6.30
GPT-4o-mini	0	1.2	∼1,500	∼$0.45
VLM Methods (Open-source, single A100)				
Qwen3-VL-30B	0	4.5	∼1,600	∼$0
Qwen3-VL-8B	0	1.8	∼1,400	∼$0

changes, VLM approaches offer significant advantages, while MARL methods may be preferred for high-frequency, cost-sensitive applications once training is completed.

4.5 Key Findings and Implications

Our comprehensive evaluation reveals several critical insights about AI decision-making paradigms in complex strategy environments:

Zero-shot vs Training Trade-off: VLM agents demonstrate remarkable zero-shot capabilities, achieving 75–81% win rates on fundamental scenarios without any training, while MARL methods struggle to achieve comparable performance even after 5M training steps with state-of-the-art Swin-Transformer backbones, confirming the extreme sample inefficiency of training spatial/tactical policies from scratch.

Cross-Modal Alignment Gap: MARL agents fail to leverage textual observations effectively (IPPO Vision+Text underperforms Vision-Only), while VLMs naturally exploit cross-modal information through pre-trained alignment. This gap highlights the unique advantage of foundation model pre-training for multimodal tactical reasoning.

Complexity Limitations: Both paradigms show performance degradation on high-complexity scenarios, but through different failure modes. MARL agents fail to learn effective coordination strategies from visual inputs, while VLM agents struggle with precise timing and micro-management despite strong strategic understanding. The 0% win rates on the most challenging scenarios across all VLMs represent genuine capability ceilings rather than engineering failures.

Human Alignment: VLM agents produce significantly more interpretable and human-like decision processes ($p < 0.01$), making them valuable for applications requiring explainable AI or human-AI collaboration.

Cost-Performance Landscape: Open-source VLMs (Qwen3-VL-30B) achieve competitive performance on simpler scenarios at zero marginal cost, while proprietary models maintain advantages on complex tasks. MARL offers fast inference after expensive training, creating complementary deployment profiles.

5 Conclusion

AVACraft establishes a standardized multimodal benchmark for cross-paradigm evaluation in StarCraft II, enabling direct comparison between training-based MARL and zero-shot VLM approaches via unified RGB, language, and structured observations. Comprehensive evaluation across 21 scenarios with six MARL algorithms and multiple VLMs (proprietary and open-source) reveals fundamental trade-offs in sample efficiency, cross-modal reasoning, interpretability, and deployment cost—MARL peaks at 19.3% after 5M steps, while VLMs achieve 75–90% zero-shot with human-aligned decisions. Beyond StarCraft II, AVACraft offers a framework for studying human-aligned AI, opening future directions including hybrid RL-VLM systems, enhanced spatial reasoning for dense formations, and scaling to full-game scenarios.

Limitations

AVACraft focuses on tactical micromanagement rather than full-game long-horizon tasks with resource and tech-tree management, and VLMs achieve 0% win rates on the most challenging maps requiring precise spatial reasoning and sustained micro-control. MARL agents may benefit from longer training budgets or curriculum strategies beyond our 5M-step evaluation. The human study, while statistically significant ($p < 0.01$), involves only seven participants. Current metrics center on win rate; finer-grained measures (APM, micro-action accuracy, token efficiency) and hybrid RL-VLM frameworks are promising directions we leave to future work.

References

Anthropic. 2024. Claude 3 model card.

Jinze Bai et al. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Anthony Brohan et al. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*.

Shaofei Cai et al. 2023. Groot: Learning to follow instructions by watching gameplay videos. *arXiv preprint arXiv:2310.08235*.

Yue Deng, Weiyu Ma, Yuxin Fan, Yin Zhang, Haifeng Zhang, and Jian Zhao. 2024a. A new approach to solving smac task: Generating decision tree code from large language models. *Preprint*, arXiv:2410.16024.

Yue Deng, Yan Yu, Weiyu Ma, Zirui Wang, Wenhui Zhu, Jian Zhao, and Yin Zhang. 2024b. Smac-hard: Enabling mixed opponent strategy script and self-play on smac. *Preprint*, arXiv:2412.17707.

Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob N. Foerster, and Shimon Whiteson. 2023. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. *Preprint*, arXiv:2212.07489.

Lei Han, Jiechao Xiong, Peng Sun, Xinghai Sun, Meng Fang, Qingwei Guo, Qiaobo Chen, Tengfei Shi, Hongsheng Yu, Xipeng Wu, et al. 2020. Tstarbot-x: An open-sourced and comprehensive study for efficient league training in starcraft ii full game. *arXiv preprint arXiv:2011.13729*.

Hongliang He et al. 2024. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*.

Ruozi Huang, Xipeng Wu, Hongsheng Yu, Zhong Fan, Haobo Fu, QIANG FU, and Yang Wei. 2023. A robust and opponent-aware league training method for starcraft ii. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zongyuan Li, Yanan Ni, Runnan Qi, Lumin Jiang, Chang Lu, Xiaojie Xu, Xiangbei Liu, Pengfei Li, Yunzheng Guo, Zhe Ma, et al. 2024. Llm-pysc2: Starcraft ii learning environment for large language models. *arXiv preprint arXiv:2411.05348*.

Haotian Liu et al. 2023. Visual instruction tuning. *NeurIPS*.

Weiyu Ma, Qirui Mi, Yongcheng Zeng, Xue Yan, Yuqiao Wu, Runji Lin, Haifeng Zhang, and Jun Wang. 2024. Large language models play starcraft ii: Benchmarks and a chain of summarization approach. *Preprint*, arXiv:2312.11865.

Michael Mathieu, Sherjil Ozair, Srivatsan Srinivasan, Caglar Gulcehre, Shangtong Zhang, Ray Jiang, Tom Le Paine, Konrad Zolna, Richard Powell, Julian Schrittwieser, et al. 2021. Starcraft ii unplugged: Large scale offline reinforcement learning. In *Deep RL Workshop NeurIPS 2021*.

OpenAI. 2023a. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

OpenAI. 2023b. Gpt-4v(ision) system card.

Davide Paglieri et al. 2025. Balrog: Benchmarking agentic llm and vlm reasoning on games. *ICLR*.

Alec Radford et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. 2019. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*.

DI star Contributors. 2021. Di-star: An open-sourse reinforcement learning framework for starcraftii. `https://github.com/opendilab/DI-star`.

Weihao Tan et al. 2024. Cradle: Empowering foundation agents towards general computer control. *arXiv preprint arXiv:2403.03186*.

Gemini Team et al. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354.

Zihao Wang et al. 2025. Jarvis-vla: Post-training large-scale vision language models to play visual games with keyboards and mouse. *arXiv preprint arXiv:2503.16365*.

Xinrun Xu et al. 2024. Mcu: An evaluation framework for open-ended game agents. *arXiv preprint arXiv:2310.08367*.

Zelai Xu, Zican Xu, Xuanhan Yi, Hongru Yuan, Xin Chen, Yi Wu, Chao Yu, and Yu Wang. 2025. Vs-bench: Evaluating vlms for strategic reasoning and decision-making in multi-agent environments. *arXiv preprint arXiv:2506.02387*.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The surprising effectiveness of PPO in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624.

Alex L. Zhang et al. 2025a. Videogamebench: Can vision-language models complete popular video games? *arXiv preprint arXiv:2505.18134*.

Chen Zhang, Qiang He, Zhou Yuan, Elvis S. Liu, Hong Wang, Jian Zhao, and Yang Wang. 2024. Advancing drl agents in commercial fighting games: Training, integration, and agent-human alignment. *Preprint*, arXiv:2406.01103.

Chen Zhang, Huan Hu, Yuan Zhou, Xu Wang, and Elvis S. Liu. 2025b. Hifas: A hybrid interactive fps agent system for large game maps. *IEEE Transactions on Games*, pages 1–13.

Zhonghan Zhao et al. 2024. Steve: See and think: Embodied agent in virtual environment. In *ECCV*.

A Impact Statement

This work advances the field of multimodal AI decision-making through the lens of real-time strategy games. While our primary contribution is methodological, we acknowledge several potential societal implications. The development of more human-aligned AI agents could enhance human-AI collaboration and improve AI system interpretability. However, advances in strategic decision-making capabilities also warrant careful consideration regarding dual-use applications. We believe our focus on human-centric design and transparent decision processes helps promote responsible AI development. Our framework primarily serves as a research tool for studying AI capabilities in controlled game environments, with minimal risk of direct negative societal impact.

B Related Work

Foundation Models for Multimodal Understanding Recent advances in Large Language Models such as GPT-4 (OpenAI, 2023a), Claude (Anthropic, 2024), and Llama (Touvron et al., 2023) have demonstrated remarkable reasoning capabilities. Building upon these foundations, Vision-Language Models (Radford et al., 2021; Liu et al., 2023) integrate visual encoders with language models, enabling simultaneous understanding of visual and textual information. Models including GPT-4V (OpenAI, 2023b), Gemini (Team et al., 2023), and Qwen-VL (Bai et al., 2023) have shown strong performance across diverse multimodal tasks, with applications spanning robotic control (Brohan et al., 2023), web navigation (He et al., 2024), and interactive environments (Tan et al., 2024).

Vision-Language Models for Game Environments Game environments have emerged as important testbeds for evaluating VLM decision-making capabilities. CRADLE (Tan et al., 2024) introduced the General Computer Control framework, demonstrating that VLMs can interact with complex AAA games like Red Dead Redemption 2 using only screenshots and keyboard-mouse actions. Minecraft has become a particularly popular platform for VLM agent research. The STEVE series (Zhao et al., 2024) combines vision models with LLMs for embodied agents capable of open-world exploration. GROOT (Cai et al., 2023) learns instruction following by watching gameplay videos without manual annotations. JARVIS-VLA (Wang et al., 2025) employs vision-language post-training for end-to-end action prediction. MCU Benchmark (Xu et al., 2024) provides a systematic evaluation framework with 3,452 atomic tasks spanning diverse skills including manipulation, navigation, and combat. Cross-game benchmarks have also been developed: BALROG (Paglieri et al., 2025) aggregates six RL environments including BabyAI, Crafter, and NetHack to evaluate long-horizon decision-making capabilities across different game genres. VideoGameBench (Zhang et al., 2025a) includes 23 classic games requiring VLMs to complete entire games using only raw visual inputs, providing insights into VLM capabilities across varied gameplay mechanics. VS-Bench (Xu et al., 2025) evaluates VLMs for strategic reasoning and decision-making in multi-agent environments across multiple games, focusing on high-level strategic capabilities. These works have demonstrated VLMs' potential for understanding game environments and generating appropriate actions based on visual observations.

StarCraft II as an AI Benchmark StarCraft II has served as a premier benchmark for artificial intelligence research, particularly for multi-agent systems requiring real-time coordination under partial observability. AlphaStar (Vinyals et al., 2019) achieved superhuman performance through a combination of imitation learning from human replays and multi-agent reinforcement learning, demonstrating that deep RL could master the game's full complexity. This work inspired numerous architectural improvements including distributed training frameworks (Mathieu et al., 2021), hierarchical decision-making (star Contributors, 2021), and macro-action abstractions (Han et al., 2020; Huang et al., 2023). For standardized multi-agent evaluation, the StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al., 2019) provided a widely-adopted framework focusing on cooperative micromanagement scenarios with decentralized execution. SMAC has facilitated significant advances in value decomposition methods, communication protocols, and credit assignment mechanisms. SMACv2 (Ellis et al., 2023) extended this foundation by introducing procedurally generated scenarios requiring adaptive closed-loop policies rather than exploiting fixed opponent behaviors. SMAC-Hard (Deng et al., 2024b) further increased tactical complexity through scenarios demanding precise ability usage and unit coordi-

nation. These benchmarks have collectively advanced multi-agent reinforcement learning research through standardized evaluation protocols and diverse tactical challenges.

Language Models for StarCraft II Decision-Making Recent works have begun exploring the integration of language models with StarCraft II environments. LLM Play SC2 (Ma et al., 2024) pioneered the application of LLMs to macro-strategic decision-making in full matches, developing the TextStarCraft II text-based environment that enables LLMs to make high-level decisions regarding resource management, unit production, and technology progression. LLM-PySC2 (Li et al., 2024) provides comprehensive access to the complete PySC2 action space along with multimodal observation interfaces including visual inputs, minimap information, and structured game state. The framework includes built-in game knowledge and example demonstrations to facilitate LLM understanding of game mechanics. LLM-SMAC (Deng et al., 2024a) demonstrates the potential of code generation paradigms for tactical decision-making by leveraging LLMs to generate decision tree code for SMAC scenarios, enabling interpretable policy representation. Additional works (Zhang et al., 2024, 2025b) have explored learning from language-based strategy descriptions and hierarchical planning. These approaches have shown that language models can understand StarCraft II's strategic and tactical concepts through textual descriptions and code generation.

While existing work has advanced both VLM-based game AI and StarCraft II research independently, current benchmarks lack unified evaluation frameworks that support both traditional MARL and modern VLM approaches for fine-grained tactical micromanagement. SMAC-family benchmarks employ abstract state representations incompatible with VLM perception, while VLM game research has primarily focused on macro-level strategies (LLM-PySC2) or cross-game evaluation (VS-Bench) rather than precise multi-unit coordination. AVACraft addresses this gap by providing multimodal observations—RGB visuals, natural language descriptions, and structured state information—within a standardized evaluation framework, enabling systematic comparison between training-based and zero-shot decision-making paradigms in tactical control scenarios.

C Limitations of Previous StarCraft II Environments

While SMAC and SMACv2 have advanced multi-agent reinforcement learning research, they have fundamental limitations for developing AI systems that can truly master StarCraft II's complex decision-making challenges:

Simplified Unit Abilities and Interactions SMAC significantly simplifies unit abilities, removing critical micro-management elements that define StarCraft II gameplay. For example, Marines and Marauders lack Stimpack abilities, Stalkers cannot Blink, and only Medivacs retain their Heal ability. This oversimplification eliminates the rich tactical depth of StarCraft II, where ability timing and targeting often determine battle outcomes. In competitive play, a Marine without Stimpack is essentially a different unit, and skilled micro-management of these abilities is central to high-level play.

Limited Unit Diversity and Compositions Both SMAC and SMACv2 feature extremely limited unit diversity, with most scenarios containing only 2-3 unit types. This fails to capture StarCraft II's emphasis on complementary unit compositions and counter strategies. For instance, the classic "Marine-Marauder-Medivac" composition requires specific control patterns that balance front-line positioning, focus fire, and healing priorities—tactical considerations absent in simplified environments.

Overly Simple Enemy AI The enemy AI in SMAC and SMACv2 follows a basic "attack spawn point" strategy without any tactical depth. It neither repositions units strategically nor prioritizes targets intelligently, creating unrealistic combat scenarios. This simplistic behavior fails to challenge agents to develop the sophisticated positioning and targeting skills needed in actual StarCraft II gameplay, resulting in strategies that don't transfer to real matches.

Abstract State Representations SMAC and SMACv2 represent the game state as abstract vectors containing unit attributes, positions, and health values, completely divorced from the visual and spatial reasoning humans use when playing. This misalignment between AI and human perception fundamentally limits the ecological validity of behaviors learned in these environments.

Questionable Randomization in SMACv2 While SMACv2 introduces procedural generation

and randomization of unit types and positions, these changes don't necessarily reflect meaningful tactical variations in StarCraft II. Random army compositions often create unrealistic scenarios that wouldn't occur in competitive play, where army composition follows strategic principles and tech progression. This randomization tests an agent's ability to handle arbitrary unit combinations but fails to evaluate tactical proficiency in realistic combat scenarios.

Focus on MARL Rather Than StarCraft II Mastery These environments were designed specifically to advance MARL algorithms rather than to develop systems that can master StarCraft II gameplay. Consequently, they prioritize properties beneficial for reinforcement learning (like simplified action spaces and reward structures) over faithful reproduction of the tactical challenges that make StarCraft II compelling.

Our AVACraft environment addresses these limitations by preserving the rich tactical depth of StarCraft II micro-management. We maintain full unit abilities, support diverse unit compositions, create realistic combat scenarios, and—most importantly—align AI perception with human gameplay experience through RGB visual inputs and natural language observations. This approach enables the development of agents that can execute sophisticated tactical maneuvers involving ability timing, positioning, and multi-unit coordination that more closely resemble human gameplay.

D Hyperparameters and Reproducibility

D.1 MARL Hyperparameters

All MARL experiments use the following configuration unless otherwise noted:

Table 9: MARL training hyperparameters.

Parameter	Value
Visual Backbone (Swin-Tiny)	
Architecture	Swin-Tiny (27.5M params)
Input resolution (screen)	160×120
Input resolution (minimap)	32×32
Feature dimension	768
Text Encoder (for Vision+Text mode)	
Model	GTE-Base
Embedding dimension	768
Fusion method	Learned linear projection
Training	
Total timesteps	5,000,000
Batch size	32 (episodes)
Learning rate	5×10^{-4}
Optimizer	Adam
Discount factor γ	0.99
ϵ-greedy (value-based)	$1.0 \rightarrow 0.05$ (linear, 50K steps)
Replay buffer size	5,000 episodes
Target update interval	200 episodes
MAPPO/IPPO Specific	
Clip parameter	0.2
GAE λ	0.95
Entropy coefficient	0.01
Number of epochs	5
Mini-batch count	1
Hardware	
GPUs	2× NVIDIA A100 40GB
Training time (3m, 5M steps)	∼65–80 hours

D.2 VLM Hyperparameters

Table 10: VLM inference hyperparameters.

Parameter	Value
Proprietary Models	
Temperature	0.7
Top-p	0.95
Max tokens (per decision)	1,024
Image resolution	1920×1080 (downscaled to API limits)
Decision frequency	2 Hz
API retry strategy	3 retries, exponential backoff
Open-source Models (Qwen3-VL)	
Temperature	0.7
Top-p	0.95
Max tokens (per decision)	1,024
Quantization	BF16
Hardware	1× NVIDIA A100 80GB
RAG Knowledge Base	
Embedding model	GTE-Base
Retrieval top-k	3
Knowledge sources	Unit stats, matchup data, pro replays

E Pseudocode

Algorithm 1 AVA Decision Pipeline for AVACraft

Input: StarCraft II environment env, History buffer size H

Initialize AVACraft environment and get initial observation $o_0 = (I_0, T_0, U_0) = env.reset()$
Initialize history buffer $\mathcal{H}$, total reward $R = 0$
while env is not terminated **do**
  // Stage 1: Micro-skill Planning
  Generate skill plan $S_t = \text{VLM}_{\text{plan}}(o_t, \mathcal{H})$
  // Stage 2: Strategic Unit Analysis
  Detect units $A_t = \text{VLM}_{\text{detect}}(I_t)$
  for each unit $u_i \in U_t$ **do**
    Parse unit info $(id_i, type_i, pos_i, attr_i, status_i)$
  end for
  Identify priority units $U_{\text{priority}} = \text{VLM}_{\text{analyze}}(o_t, S_t)$
  // Stage 3: Knowledge Integration
  for each unit $u \in U_{\text{priority}}$ **do**
    Retrieve unit knowledge $K(u) = \text{Retrieve}(type_u)$
  end for
  // Stage 4: Action Generation
  Initialize action set $a_t = \{\}$
  for each friendly unit i **do**
    if i should attack **then**
      Add $(i, j) \in \mathcal{A}_{attack}$ to a_t for target unit j
    else if i should move **then**
      Add $(i, x, y) \in \mathcal{A}_{move}$ or (i, d) to a_t
    else if i should use ability **then**
      Add $(i, \text{ability}, \text{target}) \in \mathcal{A}_{ability}$ to a_t
    end if
  end for
  // Execute action and update
  Get the reward and next observation: $r_t, o_{t+1} = env.step(a_t)$
  Update history buffer $\mathcal{H}$
  $R \leftarrow R + r_t$
  $o_t \leftarrow o_{t+1}$
  if Victory or Defeat or TimeLimit **then**
    break
  end if
end while
return total reward R

F Map Details

Our **AVACraft** environment features a diverse collection of 21 specialized maps, systematically categorized based on player count and ability usage capabilities. These maps originate from three primary sources: SMAC-based maps redesigned from the StarCraft Multi-Agent Challenge framework, original maps specifically designed for AVA evaluation, and selected scenarios adapted from the LLM-PySC2 framework[1].

Each map is meticulously designed to evaluate specific aspects of tactical proficiency and strategic decision-making:

- **Unit Control**: Assessment of fundamental micromanagement capabilities
- **Multi-Unit Coordination**: Evaluation of strategic control over heterogeneous unit compositions
- **Terrain Usage**: Testing of positional awareness and environmental exploitation
- **Kiting**: Assessment of dynamic hit-and-run tactical execution
- **Split**: Evaluation of unit distribution strategies under enemy threats
- **Ability Usage**: Testing of ability timing optimization and target prioritization

G Evaluation Metrics

We define the three metrics used in the human evaluation of AVA and MARL agents, each rated on a 1–5 scale:

- **Game Bug Exploitation**: Measures whether the agent exploits game bugs, particularly vulnerabilities in SMAC's built-in AI, which uses a flawed strategy of attacking only the enemy's spawn point and stopping if the enemy moves out of range or beyond attack distance (1 = frequent exploitation, 5 = no exploitation).
- **Reasoning Coherence**: Evaluates whether the agent's decisions are logical, incorporating StarCraft II game knowledge (e.g., unit

[1] https://github.com/NKAI-Decision-Team/LLM-PySC2

Table 11: Single player maps without ability usage.

Map Name	Unit Control	Multi Unit	Terrain Usage	Kiting	Split	Mirror Match	Units	Source
2c_vs_64zg	✓	✓	✓	✓			Player: 2 Colossi Enemy: 64 Zerglings	SMAC
2m_vs_1z	✓	✓					Player: 2 Marines Enemy: 1 Zealot	SMAC
2s_vs_1sc	✓	✓					Player: 2 Stalkers Enemy: 1 Spinecrawler	SMAC
3s_vs_3z	✓	✓					Player: 3 Stalkers Enemy: 3 Zealots	SMAC
6r_vs_8z	✓	✓	✓	✓			Player: 6 Reapers Enemy: 8 Zealots	NEW
8m1mv_vs_2st	✓	✓					Player: 8 Marines, 1 Medivac Enemy: 2 Siege Tanks	NEW
8m2st_vs_35zg4b	✓	✓	✓				Player: 8 Marines, 2 Siege Tanks Enemy: 35 Zerglings, 4 Banelings	NEW
8m_vs_2pc1wp	✓						Player: 8 Marines Enemy: 1 Warp Prism, 2 Photon Cannons	NEW
2s3z	✓	✓	✓			✓	Player: 2 Stalkers, 3 Zealots Enemy: 2 Stalkers, 3 Zealots	SMAC
3m	✓	✓				✓	Player: 3 Marines Enemy: 3 Marines	SMAC
mixed_units	✓	✓					Player: 1 Zealot, 1 Immortal, 1 Archon, 1 Stalker, 1 Phoenix Enemy: 1 Marine, 1 Marauder, 1 Reaper, 1 Hellbat, 1 Medivac, 1 Viking (Assault), 1 Ghost, 1 Banshee	NEW

matchups) and operational skills (e.g., positioning, targeting) (1 = illogical, 5 = perfect logic).

- **Human Similarity**: Assesses how closely the agent's strategies resemble human play, including techniques like hit-and-run tactics and multi-unit coordination (e.g., combined-arms strategies) (1 = unlike human, 5 = completely human-like).

Evaluations were conducted in a blinded setting: participants watched recorded replays without being informed whether the controlling agent was MARL-based or VLM-based. Each participant evaluated 10 replays (5 per paradigm) in randomized order. Statistical significance was assessed using the Mann-Whitney U test due to the ordinal nature of Likert scale data and the small sample size.

H Open-Source VLM Extended Results

To ensure reproducibility without proprietary API dependencies, we provide extended results for open-source Qwen3-VL models across all evaluated scenarios.

Table 12: Single player maps with ability usage.

Map Name	Unit Control	Multi Unit	Terrain Usage	Kiting	Split	Ability Usage	Units	Source
8m3mr1mv1st_mirror	✓	✓			✓	✓	Player: 8 Marines, 3 Marauders, 1 Medivac, 1 Siege Tank Enemy: 8 Marines, 3 Marauders, 1 Medivac, 1 Siege Tank	NEW
8s_vs_8m3mr1mv1st	✓				✓	✓	Player: 8 Stalkers Enemy: 8 Marines, 3 Marauders, 1 Medivac, 1 Siege Tank	NEW
8m3mr1mv1st_vs_5s2c	✓	✓			✓	✓	Player: 8 Marines, 3 Marauders, 1 Medivac, 1 Siege Tank Enemy: 5 Stalkers, 2 Colossi	NEW
pvz_ht	✓	✓				✓	Player: 12 Stalkers, 1 Archon, 4 Sentries, 6 High Templars Enemy: 64 Zerglings, 32 Banelings, 3 Ultralisks, 3 Queens	LLM-PYSC2

Table 15: Extended open-source VLM results. Win rates (%) over 20 episodes.

Scenario	Qwen3-VL-8B	Qwen3-VL-30B
vlm_attention_1	40	**90**
3m	40	50
2m_vs_1z	5	15
mixed_units	35	60
2s3z	10	20
3s_vs_3z	5	15
2s_vs_1sc	0	0
pvz_ht	10	20
8m2st_vs_35zg4b	15	30
8m1mv_vs_2st	0	5
8m_vs_2pc1wp	0	5
6r_vs_8z	0	0
2c_vs_64zg	0	0
Average	12.3	23.8

The clear performance jump from 8B (12.3% average) to 30B (23.8% average) demonstrates AVACraft's ability to measure VLM scaling laws in spatial reasoning tasks. Notably, Qwen3-VL-30B achieves 90% on vlm_attention_1 (a focused targeting scenario), confirming that the environment mechanics and API pipelines are entirely functional and that the 0% win rates on complex maps reflect genuine capability ceilings of current VLMs.

I Case of Study

Figures 3 and 4 illustrate the initial stages of AVA's decision-making process. The system begins by processing the raw RGB battlefield observation, then identifies and annotates individual units with their respective IDs and health status. This visual processing stage forms the foundation for subsequent tactical analysis.

Figure 3: Original RGB observation of battlefield situation in the Colossi vs Zerglings scenario.

Figure 5 demonstrates AVA's strategic decision-making capabilities. In this complex micromanagement scenario, AVA identified Zergling_52 (Tag: 54) as a priority target due to its strategic position at [2,1], where attacking it would maximize area-of-effect damage to nearby clustered units. This decision demonstrates the system's ability to not only identify low-health targets (5/35 HP) but also recognize opportunities for efficient damage distribution through Colossi's line damage mechanic. Supporting this decision, the system also identified Zergling_1 (Tag: 3) and Zergling_2 (Tag: 4) as secondary priority targets due to their threatening positions at [1,1] and [0,1] respectively, enabling a comprehensive control strategy that combines focus fire with positional advantage.

The tactical execution depicted in Figures 6, 7, and 8 showcases AVA's sophisticated decision-

Table 13: Two player maps without ability usage.

Map Name	Unit Control	Multi Unit	Terrain Usage	Kiting	Split	Mirror Match	Units	Source
MMM_vs_MMM	✓	✓		✓	✓	✓	Player 1: 8 Marines, 3 Marauders, 1 Medivac Player 2: 8 Marines, 3 Marauders, 1 Medivac	SMAC
mixed_units_pvp	✓	✓					Player 1: 1 Zealot, 1 Immortal, 1 Archon, 1 Stalker, 1 Phoenix Player 2: 1 Marine, 1 Marauder, 1 Reaper, 1 Hellbat, 1 Medivac, 1 Viking (Assault), 1 Ghost, 1 Banshee	NEW
terran_mirror	✓	✓				✓	Player 1: 1 Marine, 1 Marauder, 1 Reaper, 1 Hellbat, 1 Medivac, 1 Viking (Assault), 1 Ghost, 1 Banshee Player 2: 1 Marine, 1 Marauder, 1 Reaper, 1 Hellbat, 1 Medivac, 1 Viking (Assault), 1 Ghost, 1 Banshee	NEW

Table 14: Two player maps with ability usage.

Map Name	Unit Control	Multi Unit	Terrain Usage	Kiting	Split	Ability Usage	Units	Source
7s_vs_11m1mv1st	✓			✓	✓	✓	Player 1: 7 Stalkers Player 2: 11 Marines, 1 Medivac, 1 Siege Tank	NEW
8s_vs_8m3mr1mv1st_pvp	✓			✓	✓	✓	Player 1: 8 Stalkers Player 2: 8 Marines, 3 Marauders, 1 Medivac, 1 Siege Tank	NEW
8m3mr1mv1st_mirror_pvp	✓	✓		✓	✓	✓	Player 1: 8 Marines, 3 Marauders, 1 Medivac, 1 Siege Tank Player 2: 8 Marines, 3 Marauders, 1 Medivac, 1 Siege Tank	NEW

making processes that emerge without explicit training. The system first performs battlefield analysis, identifying Banelings as primary threats due to their splash damage potential against clustered units. It then implements a coordinated response by strategically positioning Marines at safe distances while maintaining focus fire capabilities. Throughout the engagement, AVA demonstrates multiple micro-skills simultaneously: prioritized target selection, formation control, and adaptive positioning. This behavior closely resembles human expert gameplay strategies, highlighting AVA's ability to leverage VLM reasoning for complex tactical decision-making that would typically require extensive reinforcement or imitation learning in traditional approaches.

Figure 9 illustrates AVA's ability to coordinate heterogeneous unit compositions. In the initial analysis phase (a), the system identifies critical targets including a low-health Viking Assault (11/125 HP), an energy-rich Ghost (56 energy), and support units like Medivac. Based on this assessment, it executes a coordinated attack plan (b) where each unit is assigned optimal targets: Zealot engages the weakened Viking, Phoenix provides air superiority against Medivac, Immortal focuses on armored targets, while the Archon maintains a strategic position for battlefield control. This demonstrates VLM's understanding of unit-specific attributes (health states, energy levels, armor types) and tactical synergies in mixed-unit scenarios without requiring explicit training.

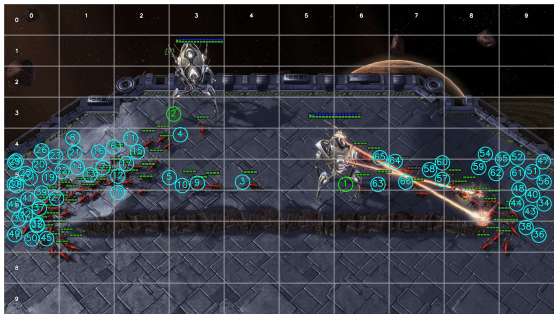


Figure 4: Annotated unit positions with unit IDs and health status.

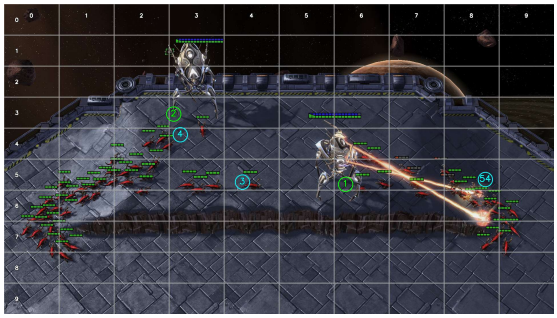
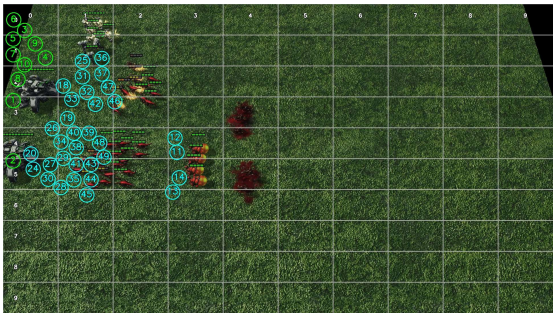


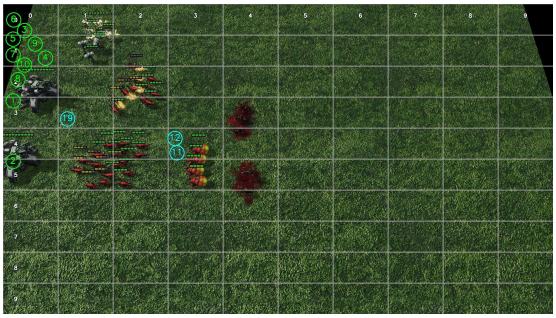
Figure 5: AVA's strategic analysis highlighting prioritized targets and optimal attack vectors.

AVA demonstrates robust performance in scenarios requiring strategic target selection and basic coordination but encounters challenges with complex micro-management tasks requiring precise ability timing (as in `2s_vs_1sc_vlm_priority`) or sophisticated terrain exploitation (as in `2c_vs_64zg_vlm_priority`, Figure 10). Through systematic analysis, we identified three primary limitations: (1) inconsistent spatial understanding in dense unit formations and (2) challenges in maintaining temporal consistency during high-frequency decision cycles.

(a) Initial state showing Marine/Tank positions



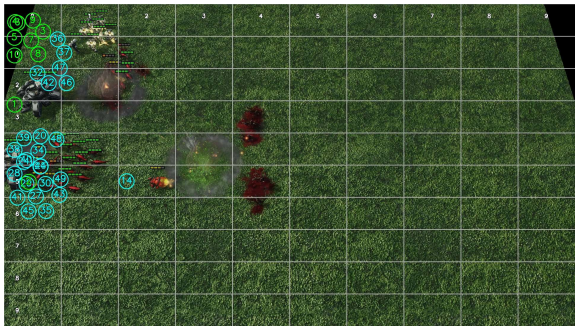
(b) VLM unit identification



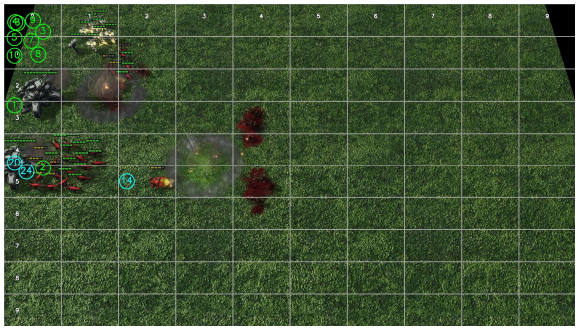
(c) Priority targeting analysis

Figure 6: Stage 1: AVA's battlefield analysis and threat assessment in Marine/Tank vs Baneling/Zergling engagement.

(a) Marine formation adjustment



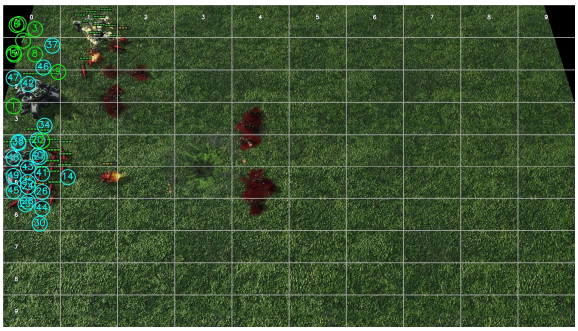
(b) Coordinated focus fire execution



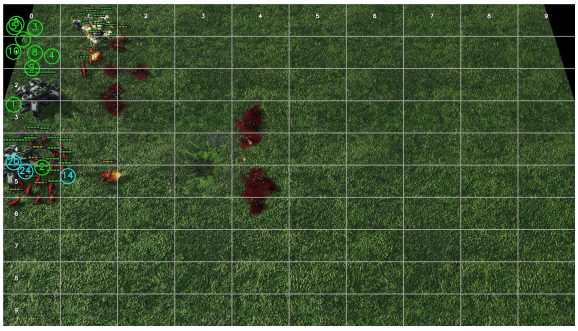
(c) Optimized Marine positioning

Figure 7: Stage 2: Tactical positioning and focus fire coordination on priority targets.

(a) Secondary target engagement



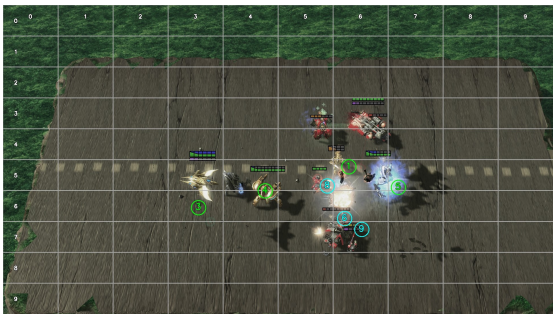
(b) Maintained spread formation



(c) Final engagement phase

Figure 8: Stage 3: Sequential target elimination while maintaining strategic formation.

(a) Initial battlefield analysis with unit annotations



(b) Coordinated attack execution and positioning

Figure 9: Multi-type unit coordination in Protoss vs Terran engagement, showing AVA's strategic targeting based on unit attributes and tactical synergies.

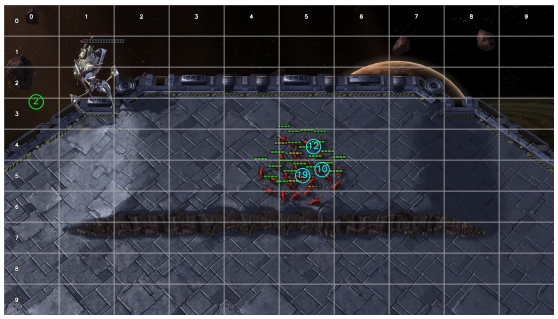


Figure 10: Tactical terrain exploitation: Colossi positioned in corner location to maximize attack range while minimizing exposure to enemy units.