Combining Spatial and Entity-Based Reasoning for Competitive MicroRTS via U-Net and Transformers

Mathis Delsart
ICTEAM, UCLouvain
Louvain-la-Neuve, Belgium

Achille Morenville
ICTEAM, UCLouvain
Louvain-la-Neuve, Belgium

Éric Piette
ICTEAM, UCLouvain
Louvain-la-Neuve, Belgium

***Abstract*—In MicroRTS, a real-time strategy benchmark, winning requires long-range coordination between many units. Yet, current deep reinforcement learning agents reason only over spatial feature maps and treat units implicitly through stacks of channels. The contributions of this paper are twofold: *(i)* `UECD`, a hybrid architecture coupling spatial and per-unit reasoning explicitly via a multi-scale convolutional backbone and a Transformer over unit entities; and *(ii)* a PPO-based training recipe derived from systematic ablation. On the *basesWorkers16x16A* map, `UECD` outranks prior competition winners, topping an open-source and reproducible tournament with a 96.67% win rate.**

***Index Terms*—Deep Reinforcement Learning, Real-Time Strategy Games, MicroRTS, Entity-Based Reasoning, Spatial Reasoning, U-Net, Transformer**

I. Introduction

Real-time strategy (RTS) games combine large combinatorial action spaces, multi-unit coordination, sparse long-horizon rewards, and real-time constraints, making them a challenging benchmark for sequential decision-making [1], [2]. AlphaStar [3] demonstrated Grandmaster-level play in StarCraft II through deep reinforcement learning (DRL), but at an extreme compute cost (384 TPUs over 44 days). MicroRTS [4] distills RTS gameplay onto small grid maps while preserving its core difficulties and has served as the IEEE Conference on Games (CoG) MicroRTS competition benchmark since 2017 [5]. Current MicroRTS agents fall into two paradigms: DRL agents such as RAISocketAI [6], dominant on small maps ($\leq 16\times16$) where per-tick inference cost remains manageable, and rule-based bots such as CoacAI, Mayari, ObiBotKenobi, and TMA, which generalize to larger layouts within tight time budgets.

Purely convolutional backbones capture long-range unit interactions only indirectly. This paper addresses this limitation through two contributions: ***(i)* `UECD`**, a hybrid architecture combining a multi-scale encoder-decoder with skip connections (U-Net) [7] gated by channel-and-spatial attention (CBAM, Convolutional Block Attention Module) [8] for spatial reasoning, and an entity Transformer [9] for unit-level reasoning; and ***(ii)* a PPO-based training recipe** [10] derived from systematic ablation, introducing *Matchup Competitiveness Weighting* and two-phase opponent-pool fine-tuning.

`UECD` is benchmarked in an open-source, reproducible tournament prompted by the 2026 CoG MicroRTS competition's shift to Large Language Model (LLM) agents. On the *basesWorkers16x16A* map, `UECD` tops the standings against past CoG winners and baselines, achieving a 96.67% win rate (WR) and a 9–1 head-to-head record against RAISocketAI. Robustness is further evaluated under layout and scale shifts.

II. Background and Related Work

MicroRTS-Py [11] exposes MicroRTS as an RL environment. Its GridNet PPO baseline, using per-cell action logits with invalid-action masking [12], reaches 91% WR on *basesWorkers16x16A* as Player 0 (P0) against a 2021-era pool. Building on this framework, RAISocketAI [6], the strongest published DRL agent for MicroRTS, trains a 5 M-parameter backbone via self-play and a fixed opponent pool; several design choices inform the ablations in Section V.

AlphaStar [3] introduced two ideas adopted here: *per-unit Transformer encoding*, which represents units as discrete entities rather than as pixels in a feature map; and *Prioritized Fictitious Self-Play* (PFSP), which samples opponents from a historical league weighted by matchup difficulty.

Further details on competition rules, mechanics, and the environment are available in [5], [11] and the project Wiki.

III. UECD Architecture

Fig. 1 shows the proposed `UECD` architecture, short for `U-Net-Entity-CBAM-Deep`. Unlike purely spatial baselines, `UECD` combines three complementary reasoning axes:

1. Spatial perception: A CBAM-gated U-Net [7], [8] backbone (Fig. 1, blue and orange) performs multi-scale spatial reasoning through skip connections between the encoder and decoder. Unlike channel-only Squeeze-and-Excitation (SE) attention [13], CBAM jointly weights channels and spatial regions (e.g. HP channels and frontlines). Most depth sits at the bottleneck, where small feature maps keep it cheap.

2. Relational reasoning: A parallel Entity Transformer [9] (Fig. 1, green) captures long-range unit interactions explicitly (e.g. units coordinating an attack). Each unit is encoded as a token combining raw attributes, spatial features, and positional embeddings. Refined entity features are scattered additively back into the spatial map at both bottleneck (strategic) and mid-resolution (tactical) scales.

3. Global spatial reasoning: A lightweight 4-head self-attention layer over the bottleneck grid (Fig. 1, red) captures long-range region-to-region dependencies beyond local convolutional receptive fields (e.g. front push vs. base defense).

979-8-3315-9476-3/26/$31.00 ©2026 IEEE

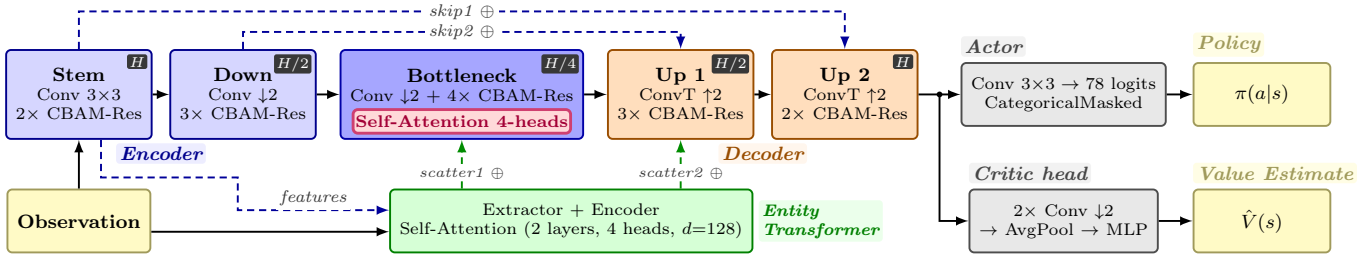

Fig. 1. **UECD Architecture:** CBAM-gated U-Net encoder-decoder (blue/orange) with bottleneck self-attention (red) and a parallel Entity Transformer (green), scattering enriched features at two scales. Actor and critic heads (gray) produce policy logits $\pi(a|s)$ and value estimates $\hat{V}(s)$ from the observation (yellow).

TABLE I
ARCHITECTURE ABLATION (3 SEEDS, 100 M STEPS EACH).

Architecture	Params	SPS	Mean ± Std	$\bar{\Delta}$
GridNet	838 k	2 688	64.8 ± 3.8	N/A
IMPALA-CNN	1.02 M	2 416	68.7 ± 5.8	+3.9
IMPALA-CNN-Entity	1.37 M	1 903	74.9 ± 1.2	+10.1
U-Net	2.59 M	2 121	77.0 ± 1.6	+12.2
U-Net-Entity	2.90 M	1 518	82.7 ± 4.8	+17.9
U-Net-Entity-CBAM	2.90 M	1 289	79.1 ± 7.2	+14.3
U-Net-Entity-CBAM-Deep	4.72 M	1 176	**86.3 ± 7.0**	**+21.5**

TABLE II
FEATURE ABLATION (3 SEEDS, 50 M STEPS EACH).

Feature	Mean ± Std	$\bar{\Delta}$
Extended Observation (73 channels) [6]	84.3 ± 1.7	+26.3
Filtered Masks + Reserved Observation [6]	80.6 ± 5.6	+22.6
Matchup Competitiveness Weighting (MCW)	78.4 ± 0.4	+20.4
Opponent Modeling [16]	77.8 ± 4.0	+19.8
Partial Advantage Estimation (PAE, 95%) [17]	71.3 ± 8.1	+13.3
Adaptive opponent curriculum	69.6 ± 1.8	+11.7
Adaptive value norm. (PopArt) [18]	68.5 ± 8.7	+10.5
Mixed bot + PFSP self-play [3]	66.9 ± 9.1	+8.9
Gaussian Error Linear Units (GELU) [19]	66.3 ± 8.2	+8.3
Frame stacking (4 frames) [20]	65.3 ± 4.5	+7.3
Autoregressive action sampling [3]	63.7 ± 7.6	+5.7
Spatial pyramid pool (SPP) critic [21]	63.2 ± 8.3	+5.2
Baseline	58.0 ± 7.5	N/A

IV. Setup and Methodology

Training runs on the standard 16×16 *basesWorkers16x16A* map (4000-step episode cap), across 24 parallel bot environments (9 CoacAI, 9 Mayari, 2 RandomBiasedAI, 2 POWorkerRush, 2 POLightRush; sides swapped per episode). PPO [10] with Generalized Advantage Estimation (GAE) [14] uses γ=0.99, λ_{GAE}=0.95, ϵ_{clip}=0.1, rollouts/epochs/minibatches = 512/2/3, c_v=0.5, max grad norm 0.5; learning rate and entropy linearly annealed from $2.5e{-}4 \to 0$ and $1e{-}2 \to 1e{-}3$, respectively. Reward weights follow [11].

Game counts are split evenly between P0 and P1. WRs are averages over 1000 stochastic games per opponent (100 in Fig. 6). The tournament (Section VI) uses the deterministic MicroRTS CoG protocol (10 games per matchup). Architecture and feature ablations (Sections V-A–V-B) use 3 seeds.

V. Experiments and Results

A. Architecture Ablation

The three backbones are GridNet, IMPALA-CNN [15] (a deeper residual CNN), and U-Net. Each addition trades more parameters and lower SPS (Steps Per Second) for a higher WR gain ($\bar{\Delta}$). Each architecture is trained for 100 M steps over 3 seeds (21 runs, ≈ 13.6 GPU-days). Table I confirms Section III's three axes. ***(i)*** U-Net trends above IMPALA-CNN by +8.3%, supporting multi-scale spatial perception. ***(ii)*** The Entity Transformer lifts both backbones (+6.2% IMPALA-CNN, +5.7% U-Net), since RTS interactions are naturally entity-level rather than cell-level, and CNNs encode them only indirectly through stacked local fields. ***(iii)*** CBAM alone shows no clear gain (−3.6%); benefits emerge only with *-Deep* (more capacity), where UECD reaches 86.3% WR ($\bar{\Delta}$ = +21.5%).

B. Feature Ablation

Table II reports each feature's WR gain ($\bar{\Delta}$) over the UECD baseline when added in isolation (50 M steps, 3 seeds, 39 runs, ≈19.4 GPU-days), capturing main effects but not interactions.

The ranking shows a sharp discontinuity at the fourth feature: the top-4 yield ≥+19.8% (≤5.6% std), while the fifth drops to +13.3% (8.1% std), with the mean gain falling and stds elevated thereafter. The dichotomy is clean: each top-4 feature enriches the *training signal* (Extended Observation [6] adds environment channels; Filtered Masks + Reserved Observation [6] tighten action validity; MCW concentrates gradients on informative matchups; Opponent Modeling [16] adds an auxiliary opponent-prediction head). The remaining eight tune *optimization* (PAE [17], PopArt [18], GELU [19]), *inductive bias* (autoregressive action sampling [3], SPP critic [21]), or add a *redundant signal* already saturated by the top-4 (adaptive curriculum, frame stacking [20]). PFSP self-play [3] is deferred until bots saturate (Section V-C).

Of the 12 features, only MCW is novel; the rest draw from prior work. This gradient-reweighting scheme scales each PPO sample by a weight peaking at 1.0 against 50%-WR opponents (highest learning signal) and flooring at 0.5 against extreme-WR opponents (saturated, no signal). Unlike sampling-based methods (PFSP [3]) or off-policy reweighting (Prioritized Experience Replay, PER [22]), MCW reweights gradients on-policy without changing the sampling distribution.

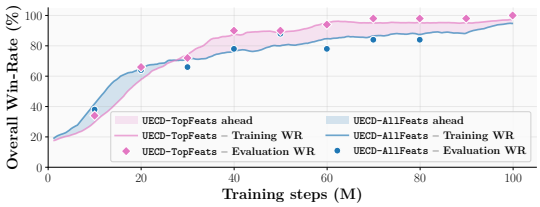

Fig. 2. **Training Curves:** WRs of `UECD-TopFeats` vs `UECD-AllFeats`.

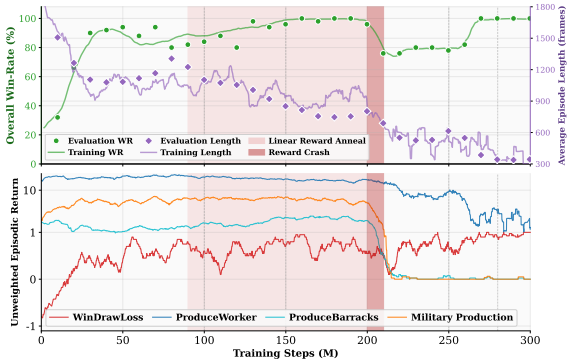

Fig. 3. **Rush Collapse:** WR (green) and episode length (purple); unweighted per-component returns (bottom); linear reward anneal (pink/red zones).

Fig. 2 validates the top-4 cutoff: with 40% less compute, `UECD-TopFeats` pulls ahead of `UECD-AllFeats` on overall pool WR after ∼20–30 M steps and converges to the same plateau, confirming that the extra eight features inflate training cost without lifting final performance.

C. Final Training Run

*1) **Rush Collapse Under Reward Anneal**:* Fig. 3 shows a training recipe (top-4 features, 300 M steps, shaped reward annealed linearly over [90 M, 210 M], PFSP self-play) that collapses into a rush strategy as shaping fades. During the anneal, WR (top, green) climbs toward 100% while episode length (top, purple) drifts from ∼1200 to ∼800 frames. The crash happens in the final 10 M: WR drops to ∼75% (recovering by 270 M) and episode length to ∼300 frames. Per-component returns (bottom) confirm the worker rush overfit: only *WinDrawLoss* keeps rising, while *ProduceBarracks* and *MilitaryProduction* fall to zero, and *ProduceWorker* drops from ∼13.0 to ∼3.0. This discount-induced shortcut is the optimal policy. Under terminal-only rewards, the RL objective $\mathbb{E}\left[\sum_{t=0}^{T} w_t^{\mathrm{T}} R_t \gamma^t\right]$ collapses to $\mathbb{E}\left[w_T R_T \gamma^T\right]$, which for consistently-winning policies can be written as $w_T R_T \mathbb{E}\left[\gamma^T\right]$. This objective is maximized by minimizing T since $\gamma < 1$, biasing fast wins over robust play. PPO's advantage-based updates inherit the same bias. With γ=0.99, shortening T from 750 to 300 amplifies the terminal signal by $\gamma^{300}/\gamma^{750} \approx 90$. On unseen bots, WR collapses from 56.5% to 0% (TMA) and 70.4% to 1.5% (ObiBotKenobi) between 150 M and 300 M.

*2) **Opponent-Pool Fine-Tuning**:* To address this, training resumes from the 150 M checkpoint with shaped weight floored at 0.1 rather than 0.0, preserving a dense productive-

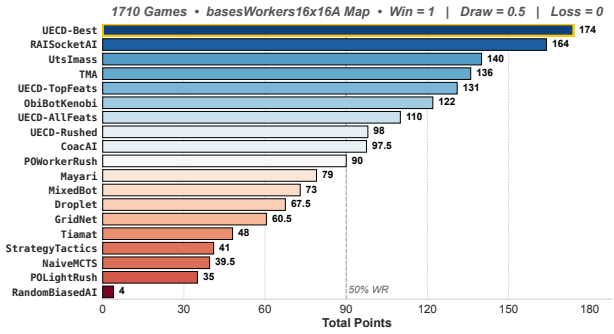

Fig. 4. **Final Tournament Standings:** 19 agents ranked by total points.

action signal. Two fine-tuning phases face progressively stronger opponents. In Phase 1, TMA and ObiBotKenobi (2 envs each) replace the three weakest bots; self-play rises from 12 to 16 envs (8 latest / 8 PFSP-hard; 24 total envs). In Phase 2, RAISocketAI joins at 4 envs, the rest at 1 env each; self-play unchanged (24 bot envs). At 150 M, learning rate, entropy, and shaped weight start at ($5e{-}5$, $5e{-}3$, 1.0); they anneal linearly to ($2e{-}5$, $2e{-}3$, 0.25) at 240 M (end of Phase 1), and to ($9.5e{-}6$, $1.2e{-}3$, 0.10) at 350 M (end of Phase 2).

The resulting agent, `UECD-Best`, is evaluated in Section VI. Training required **9.4 GPU-days** on an RTX 6000 Ada (48 GB) across **350 M steps**, roughly 7× cheaper than RAISocketAI's 70 GPU-days / 1.5 B steps over 8 maps [6], trading breadth for depth through single-map specialization.

VI. Round-Robin Tournament

The 19-agent tournament[1] evaluates 15 external agents (baselines, past CoG winners) and 4 UECD checkpoints under the deterministic MicroRTS CoG protocol (Section IV).

`UECD-Best` tops the standings with **174 points** (Fig. 4; per-game scoring: Win = +1, Draw = +0.5, Loss = +0), ahead of RAISocketAI (164 points) and every intermediate `UECD` checkpoint. Rank-1 holds excluding these, ruling out a within-family bias. In head-to-head (Fig. 5), `UECD-Best` wins **100%** against all except RAISocketAI (90%) and POWorkerRush (50%). However, under the stochastic protocol, `UECD-Best` wins 99.3% against POWorkerRush, resolving the deterministic-protocol artifact specific to this matchup. Against RAISocketAI, the WR drops to 65.7%, a narrower but still decisive margin. The improvement from `UECD-TopFeats` (131) to `UECD-Best` (174) confirms that two-phase fine-tuning bridges the gap to competitive play.

`UECD-Best`'s replays reveal a narrow Ranged-only composition with tight micro-management. Game-theoretic analyses of the 19×19 WR matrix (Copeland, α-Rank, and average regret) also rank `UECD-Best` first, with RAISocketAI second.

VII. Discussion and Limitations

`UECD-Best`'s single-map specialization isolates architectural and training contributions for clean per-component sig-

[1]**Supplementary Website:** 36 recorded `UECD-Best` matchups, extended tournament game-theoretic metrics, source code, and CoG bots archive. Available at https://mathisdelsart.github.io/microrts-drl-uecd-website/.

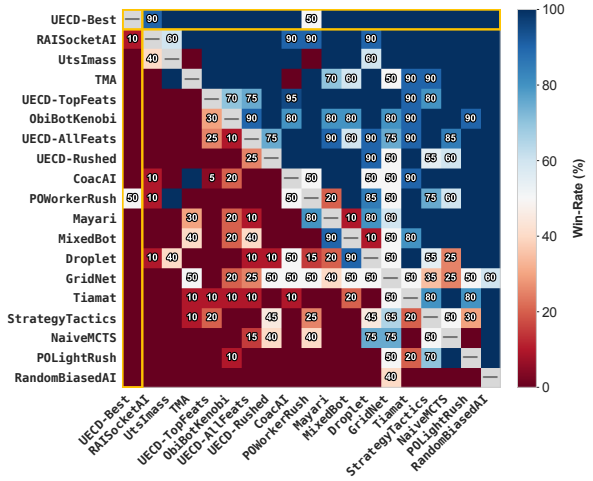

Fig. 5. **Head-to-head WR Matrix:** WR_{ij} = row agent i vs column agent j

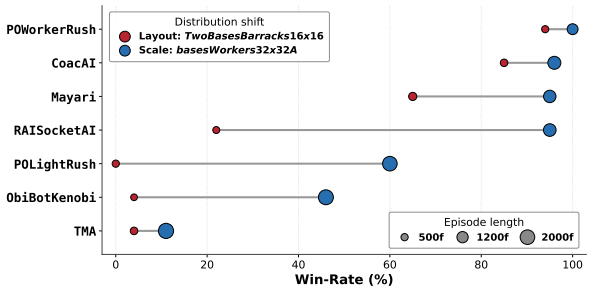

Fig. 6. **`UECD-Best` Generalization Probes:** *layout* vs *scale* shifts.

nal. Two distribution-shift probes assess the resulting out-of-distribution robustness (Fig. 6): a ***layout* shift** (same size, different topology) and a ***scale* shift** (same layout, larger size).

The asymmetry is striking: scale-shift WR sits well above layout-shift WR for every opponent, and the gap widens against the strongest agents. Episode lengths, however, stay comparable across opponents in both shifts: the policy is not disoriented under distribution shift, just sub-optimal. The learned representation encodes map-specific geometric and topological priors rather than transferable strategic invariants. These findings only describe `UECD-Best` on shifts of its training map; performance under multi-map training or on unseen maps remains an open question (Section VIII).

VIII. Conclusion and Future Work

`UECD`'s explicit decomposition of RTS reasoning (spatial, relational, global; Section V-A) beats purely-spatial backbones at modest compute. The top-4 features (Section V-B) and training recipe (Section V-C) elevate `UECD-Best` to first place in the 19-agent tournament on *basesWorkers16x16A* (Section VI) with a **96.67%** WR and a **9–1** head-to-head record against RAISocketAI on a **9.4** GPU-day budget.

The architecture is already size-agnostic; future work targets joint training on small maps ($\leq 16\times16$) via padded environments and Prioritized Level Replay (PLR) [23], then scaling to 64×64, where per-unit reasoning matters most.

Acknowledgment

Computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region.

Anthropic's Claude (Opus 4) was used for English language polishing of the paper.

References

[1] M. Buro, "Real-time strategy games: A new AI research challenge," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2003, pp. 1534–1535.

[2] S. Ontañón *et al.*, "A survey of real-time strategy game AI research and competition in StarCraft," *IEEE Trans. Comput. Intell. AI Games*, vol. 5, no. 4, pp. 293–311, Dec. 2013.

[3] O. Vinyals *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, Nov. 2019.

[4] S. Ontañón, "The combinatorial multi-armed bandit problem and its application to real-time strategy games," in *Proc. AAAI Conf. Artif. Intell. Interact. Digit. Entertain. (AIIDE)*, 2013, pp. 58–64.

[5] S. Ontañón, N. A. Barriga, C. R. Silva, R. O. Moraes, and L. H. S. Lelis, "The first MicroRTS artificial intelligence competition," *AI Mag.*, vol. 39, no. 1, pp. 75–83, Mar. 2018.

[6] S. Goodfriend, "A competition winning deep reinforcement learning agent in microRTS," in *Proc. IEEE Conf. Games (CoG)*, 2024, pp. 1–8.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Interv. (MICCAI)*, 2015, pp. 234–241.

[8] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[9] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5998–6008.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, arXiv:1707.06347. [Online]. Available: https://arxiv.org/abs/1707.06347

[11] S. Huang, S. Ontañón, C. Bamford, and L. Grela, "Gym-μRTS: Toward affordable full game real-time strategy games research with deep reinforcement learning," in *Proc. IEEE Conf. Games (CoG)*, 2021, pp. 1–8.

[12] S. Huang and S. Ontañón, "A closer look at invalid action masking in policy gradient algorithms," in *Proc. Int. FLAIRS Conf.*, 2022.

[13] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 7132–7141.

[14] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016.

[15] L. Espeholt *et al.*, "IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 1407–1416.

[16] S. Manandhar and B. Banerjee, "Reinforcement actor-critic learning as a rehearsal in MicroRTS," *Knowl. Eng. Rev.*, vol. 39, p. e6, Nov. 2024.

[17] Y. Jin, X. Song, G. Slabaugh, and S. Lucas, "Partial advantage estimator for proximal policy optimization," *IEEE Trans. Games*, vol. 17, no. 1, pp. 158–166, Mar. 2025.

[18] H. van Hasselt, A. Guez, M. Hessel, V. Mnih, and D. Silver, "Learning values across many orders of magnitude," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2016, pp. 4287–4295.

[19] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, arXiv:1606.08415. [Online]. Available: https://arxiv.org/abs/1606.08415

[20] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[22] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016.

[23] M. Jiang, E. Grefenstette, and T. Rocktäschel, "Prioritized level replay," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 4940–4950.