Can AI Learn to Play like a Pro?: A Case Study on using Transformers for StarCraft II

Prannoy Namala
Department of Mechanical Engineering
University of Maryland
College Park, MS, USA
pnamala@umd.edu

Jeffrey W. Herrmann
Department of Mechanical Engineering
Catholic University of America
Washington, DC, USA
herrmannj@cua.edu

I. Introduction

Learning coordinated behaviors in multi-agent systems is a core challenge in robotics, control, and reinforcement learning. Video-game testbeds, particularly the StarCraft Multi-Agent Challenge (SMAC) and the recently released SMACv2 [1], have become widely used environments for evaluating cooperative multi-agent reinforcement learning (MARL). Their complex, partially observable combat encounters require spatial reasoning, teamwork, and adaptation to adversaries, making them ideal for studying decentralized control.

However, most work in these environments relies on synthetic experience generated by MARL methods rather than expert demonstrations. Meanwhile, thousands of professional StarCraft II replay files capture how human players coordinate, assign roles, and switch strategies across dynamic contexts. Despite this abundant and high-quality dataset, no current work systematically leverages replay data to infer and learn strategic coordination behaviors. Such data could provide a powerful foundation for building interpretable, data-driven models of multi-agent strategy, linking human-level decision-making with control-theoretic formulations.

This work introduces a data-driven framework for inferring and imitating multi-agent strategies from professional tournaments, integrating Hamilton–Jacobi (HJ) reachability analysis with graph-based imitation learning. The objective is to bridge the gap between control-theoretic structure and learning-based adaptability, enabling the discovery of interpretable, transferable behaviors that generalize across multi-agent coordination scenarios. We develop a two-stage framework:

1) A strategy inference pipeline that automatically converts replay trajectories into labeled strategic segments using geometric descriptors and HJ-inspired templates.
2) A learning framework that trains a centralized expert model grounded in reachability concepts and distills its policy into a decentralized graph-based RL agent.

This work advances AI-driven control by demonstrating how control-theoretic abstractions can guide data-driven imitation in dynamic and uncertain environments.

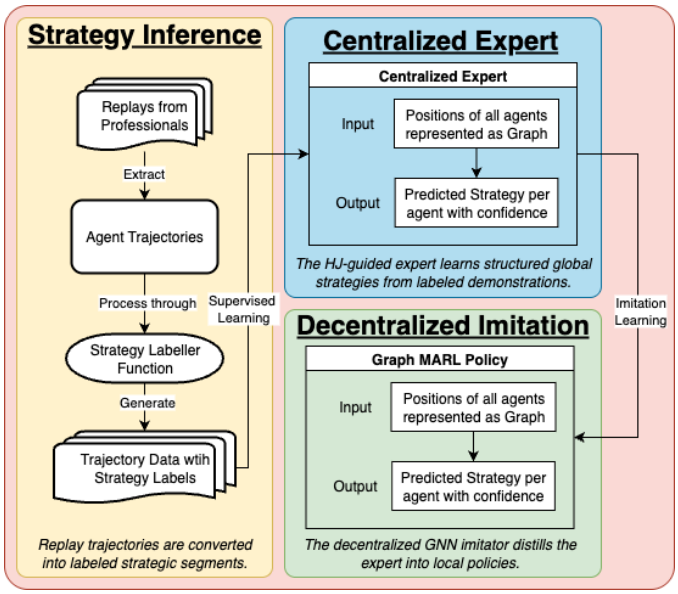

Fig. 1. Overview of the proposed pipeline. Raw StarCraft II replays are converted into labeled strategic behaviors using geometric features (Stage 1). A Hamilton–Jacobi–guided centralized expert learns structured strategy representations from these labels (Stage 2). Finally, a decentralized graph neural network imitator distills the expert's policy into local agent controllers (Stage 3).

II. Methodology

A. Strategy Inference

The professional StarCraft II games are archived as SC2Replay files. These files contain the positions of all the agents deployed and the events in the game. We develop a pipeline that automatically infers multi-agent strategies from SC2Replay files.

We first parse the SC2Replay files to extract the positions, ownership, and lifetimes of all agents across time. Each replay is converted into a unified trajectory dataset containing the agent type, timestep, position, team (ally/enemy), and status (alive/dead). Using moving window analysis, we identify 1 vs 1 (one agent from one team vs one from another) and 2 vs 1 (two agents from one team vs one from another) agent groups for each window over the trajectory. For each of the agent groups, we computed geometric features inspired by HJ

reachability. Each window's features are matched against a library of HJ-based strategy templates, producing interpretable labels with confidence levels. We call this process the Strategy Labeling Function (see Fig. 1). After we get the labels for each window, adjacent windows with identical labels are merged into continuous segments, averaging confidences and discarding shorter segments. The resulting dataset is a compact, time-aligned annotation of replay trajectories with discrete strategic labels.

B. Policy Learning & Decentralization

Using the labeled data, a centralized expert model that predicts the optimal strategic mode for each agent group based on their joint state. The model, referred to as the centralized expert from here on, is implemented as a multi-head transformer that encodes spatial-temporal dependencies among all agents and outputs a softmax over the list of strategies. The centralized expert model aims to learn a rich, globally consistent representation of multi-agent strategy from expert data. The input to this model is the joint system state, represented as a graph, where nodes correspond to agents and edges capture relative spatial or contextual relationships. The model is trained to map these joint graph states to higher-level strategic actions or distributions that reflect the behaviors identified during the strategy inference stage. While the specific architecture remains flexible, the model will integrate principles from Hamilton–Jacobi reachability to encode geometric feasibility and safety, ensuring that the resulting strategies remain interpretable and dynamically consistent. The learned expert policy serves as a centralized controller that captures both the global coordination patterns and the reachability structure underlying expert demonstrations.

To enable scalable and distributed execution, we will train a decentralized, graph-based policy to imitate the centralized expert. This model also operates over graph-structured states, allowing each agent to reason about local information through its neighborhood connections. The decentralized policy learns to reproduce the expert's strategic outputs using local observations and limited communication, enabling robust coordination in larger or dynamic teams. We will explore various imitation learning formulations to determine which best transfers the expert's global knowledge into decentralized agent behaviors.

III. Proposed Experiments and Evaluation

The proposed framework will be evaluated within the StarCraft Multi-Agent Challenge (SMAC) environment [1]. Representative scenarios such as 3m, 8m, and MMM2 will be used to benchmark both the centralized and decentralized components of the model. Evaluation will be focussed on:

- *Win Rate Against Baselines*: The principal metric will be the average win rate across multiple evaluation maps. We will measure how frequently the learned policy outperforms baseline controllers under identical environment setups and training budgets.
- *Robustness and Generalization*: Robustness will be assessed by testing policies under unseen configurations such as altered initial agent placements, different team ratios, and varied map layouts. The hypothesis is that embedding reachability-based geometric constraints will yield policies that maintain performance under these perturbations, demonstrating greater resilience than purely learned baselines like MAPPO.
- *Sample Efficiency and Training Stability*: Training curves will be analyzed to compare convergence speed and variance across runs. Because the proposed model leverages expert-labeled data rather than exploration-heavy RL, we anticipate faster convergence and more stable learning dynamics.

We expect the HJ-guided expert to outperform model-free baselines in terms of win rate consistency and interpretability, serving as a strong supervisory signal for imitation learning. The graph-based decentralized imitator is anticipated to retain this performance advantage while improving scalability and robustness, demonstrating competitive or superior results to MAPPO under dynamic or partially observed conditions.

IV. Discussion and Future Work

This work aims to improve multi-agent coordination by combining control-theoretic structures with graph-based learning. HJ reachability provides geometric priors that help the centralized expert learn stable, safe strategies that generalize beyond specific maps—addressing overfitting issues seen in model-free methods such as MAPPO. A decentralized graph imitator then distills this expert knowledge into a scalable policy, using local message passing to preserve coordinated behavior. This structure provides robustness against agent dropout, communication limitations, and environmental noise, outperforming conventional MARL approaches in dynamic settings.

Beyond StarCraft II, the same methodology can be extended to team-based physical domains where rich trajectory and tactical data are available. For example, soccer, basketball, and American football provide high-resolution positional data from tactical-view camera systems that record professional gameplay. By applying the same replay-to-strategy inference and imitation pipeline, one could extract strategic primitives such as formations, passing lanes, zone coverage and train reachability-grounded graph policies that emulate expert coordination. Such a cross-domain generalization would illustrate how the proposed approach can bridge game analytics, multi-agent reasoning, and robotic control under a unified modeling framework.

Ultimately, this line of work aims to bridge formal control theory, imitation learning, and large-scale behavioral modeling, paving the way for a new generation of interpretable, general-purpose multi-agent control systems applicable across simulation, robotics, and human team domains.

References

[1] Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob Nicolaus Foerster, and Shimon

Whiteson. SMACv2: An improved benchmark for cooperative multi-agent reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.