# HORSES FOR COURSES: ROLE-CONFIGURATION-DRIVEN LONG-HORIZON STABLE CONTROL FOR MULTI-AGENT SYSTEMS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Long-horizon stable control in multi-agent systems requires agents to maintain task performance while reacting to capability changes, disturbance propagation, and evolving system stability. Agents' functional contributions may change over time, requiring the controller to adapt its control emphasis according to the current role configuration of the team. We propose `RoleFit`, a role-configuration-driven method for long-horizon stable multi-agent control. `RoleFit` first constructs structured role evidence from local observations, executed actions, and coupled system conditions, and infers each agent's functional role as a Primary Contributor, Compensatory Contributor, or Limited Contributor. It then builds team-level role-topology evidence to reason about the current control mode: Balanced Control, Redistribution Control, or Stabilization Control. The selected mode is executed by a corresponding homogeneous role-conditioned policy, so that language-model reasoning is used for structured role and mode decisions, while low-level actions remain generated by trained control policies. Experiments on power-grid voltage control, traffic signal control, and collaborative robotic transport show that `RoleFit` improves long-horizon stability, reduces unsafe failures, and achieves more robust performance under changing disturbances.

## 1 INTRODUCTION

Multi-agent systems are increasingly deployed in long-horizon control scenarios where stable operation must be maintained over extended periods Wang et al. (2022); Li et al. (2025). In power grids, agent-based control units continuously regulate voltages and power flows under fluctuating solar and wind generation, time-varying loads, and local equipment constraints. A small mismatch between generation, load, and regulation capacity may first appear as a local voltage deviation, but can further trigger protection actions, cascading disconnections, or large-scale service interruptions when the system cannot recover in time. Investigations into the 2025 Iberian blackout pointed to a voltage surge as an immediate cause, and European grid operators later called for stronger control of voltage swings after the outage Sabolić et al. (2025). These scenarios require agents to sustain coordinated behavior over long horizons, rather than merely complete a short-term task or maximize episodic reward Berducci et al. (2024).

Many studies on multi-agent reinforcement learning have explored how agents can coordinate through centralized training with decentralized execution, value decomposition, explicit communication, parameter sharing, graph-based policies, and safety-constrained optimization Yu et al. (2022); Zhong et al. (2024). In power-grid control, recent work formulates voltage regulation or dispatch as a multi-agent decision process Chen et al. (2023), where distributed controllers learn to adjust reactive power, dispatch resources, or respond to renewable fluctuations under shared system-level objectives. Some studies further incorporate safety constraints Yan & Xu (2024); Wei et al. (2025), spatial-temporal graph information, or robust control objectives to reduce voltage violations and improve renewable accommodation. These designs provide useful mechanisms for learning coordinated actions, but they usually focus on optimizing a fixed control objective under the current observations. In long-horizon stable control, the system may require different control emphases as local regulation margins change, disturbances propagate, and some agents become less capable of



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

contributing effectively. Therefore, studying how the functional configuration of agents affects the choice of control mode can help the system maintain stability under changing operating conditions.

In this paper, we focus on functional role perception in long-horizon multi-agent stable control. In a distribution grid, one node may provide major reactive power support, another may compensate voltage deviations in a neighboring area, while a third node may have limited regulation margin because its local capacity is close to saturation. These agents are not merely in different local states; they play different functional roles in the system-level regulation process. When sufficient contributors are available, the system can emphasize normal voltage regulation and control efficiency. When one contributor becomes limited, the surrounding agents may need to redistribute regulation effort to prevent the deviation from spreading. When multiple contributors are limited or the voltage deviation grows rapidly, the system should shift toward safety-priority stabilization, even at the cost of higher control effort or lower short-term efficiency. This suggests that the role configuration of agents provides structured information about what kind of control response the system currently needs.

Designing a role-aware control method raises three challenges. First, the roles should describe functional contribution rather than local health states. A useful role representation should distinguish agents that provide primary task contribution, agents that compensate system imbalance, and agents whose effective contribution is currently limited. Second, agent-level roles should be lifted to a system-level control decision. The same set of agent roles may require different control modes when agents with constrained contribution appear in different topological positions or when the stability condition evolves differently. Thus, the method needs to reason over role distribution, interaction topology, and stability context together. Third, the selected control mode should be executed through stable low-level policies.

Inspired by the "Horses for Courses" principle, we propose `RoleFit`, a role-configuration-driven long-horizon stable control method for multi-agent systems under changing operating conditions. The core idea is to use the current functional role configuration to select a suitable control mode, and then execute the corresponding homogeneous role-conditioned policy. `RoleFit` consists of three components. First, *Functional Role Identification* characterizes agents as Primary Contributors, Compensatory Contributors, or Limited Contributors based on structured evidence from local observations, executed actions, and coupled system conditions. Second, *Control Mode Reasoning* constructs team-level role-topology evidence and infers whether the system should use Balanced Control, Redistribution Control, or Stabilization Control. Third, *Mode-Guided Policy Switching* trains one shared policy for each control mode and switches among these policies with simple stability-oriented constraints. Through this design, `RoleFit` uses language-model reasoning for structured role and mode decisions, while keeping low-level action generation in trained homogeneous control policies.

We evaluate `RoleFit` in three representative long-horizon control domains: distributed power-grid voltage control Wang et al. (2021b), traffic signal control Krajzewicz et al. (2012); Ault & Sharon (2021), and collaborative robotic transport Gupta et al. (2017); Terry et al. (2021). In each domain, we consider changing operating conditions that affect agent contribution over time, including reduced regulation capacity in power-grid nodes, increased traffic demand at selected intersections, and actuator degradation in collaborative walkers. These changing conditions require the system to identify the functional roles of agents and select an appropriate control response, ranging from normal control to load redistribution and safety-priority stabilization. We compare `RoleFit` with single-policy multi-agent reinforcement learning, mixed-training policies that are exposed to all disturbance types, and rule-based switching variants. We assess both task performance and stability using metrics such as long-term success rate and stability rate. These experiments are designed to examine whether role-configuration-driven control can improve long-horizon stability beyond a universal policy trained for all situations.

The contributions of this paper are as follows:

- We introduce a role-configuration perspective for long-horizon stable multi-agent control, where agent functional roles and their team-level configurations are considered as key factors for adaptive control.

- We propose `RoleFit`, a role-configuration-driven stable control method that identifies functional roles, reasons about system-level control modes, and switches among homogeneous role-conditioned policies under changing operating conditions.
- We evaluate `RoleFit` on three long-horizon multi-agent control tasks, including power-grid voltage control, traffic signal control, and collaborative robotic transport. The results demonstrate improved task success and long-term stability under diverse disturbances.

## 2 Related Work

### 2.1 Stability in Multi-Agent Systems

Classical multi-agent control studies stability through consensus, synchronization, formation control, and Lyapunov-based analysis Ren & Beard (2005); Olfati-Saber et al. (2007); Qin et al. (2016). These methods usually provide formal stability guarantees under structured system dynamics, known interaction models, or carefully designed control laws. Recent multi-agent reinforcement learning methods study stability from a learning-based perspective. In cooperative control tasks, methods based on centralized training with decentralized execution learn policies that optimize shared rewards while allowing each agent to act from local observations Foerster et al. (2018); Guo et al. (2025). Value-decomposition methods factorize the team value function into agent-wise utilities and are widely used in cooperative decision-making Wang et al. (2021a); Li et al. (2024). Policy-gradient method MAPPO, learned decentralized policies with a centralized critic and showed strong empirical performance in cooperative multi-agent benchmarks Yu et al. (2022); Zhong et al. (2024). Several works further incorporate safety or stability considerations into reinforcement learning. Lyapunov-based reinforcement learning used a Lyapunov function to constrain policy updates and reduce unsafe behaviors Zhang et al. (2020). Constrained policy optimization and safe exploration methods introduced explicit constraints on unsafe states, energy usage, or system violations Gu et al. (2024). In power-grid control, some methods design reward functions or safety layers to reduce voltage violations and maintain the system within operational limits Yan & Xu (2024); Wu et al. (2025). Instead of only adding stability penalties to a single policy, we consider how the functional role configuration of agents changes the control demand of the system.

### 2.2 Role-Based Multi-Agent Reinforcement Learning

Role-based multi-agent reinforcement learning studies how agents can form different behavioral patterns or functional responsibilities during cooperation. Early role-based coordination methods often define roles manually or assign roles according to prior task knowledge Wilson et al. (2010); Le et al. (2017). In learning-based settings, roles are usually learned from trajectories, action preferences, local observations, or latent representations. ROMA learns emergent roles by encouraging agents with similar roles to produce similar behaviors and agents with different roles to become distinguishable Wang et al. (2020a). It introduces a role encoder and regularization terms to organize latent roles in cooperative multi-agent reinforcement learning. RODE further uses roles to decompose the action space and learn role-conditioned policies, so that each role focuses on a subset of actions relevant to the current coordination task Wang et al. (2020b). Other methods learn dynamic roles from temporal contexts, infer roles through graph structures, or use role assignments to improve scalability in large multi-agent systems Cheng et al. (2026); Goel et al. (2025); Gao et al. (2026). Different from existing role-based MARL methods, this paper does not use roles mainly to decompose the action space or assign fixed agent identities. We define roles according to the current functional contribution of agents in long-horizon stable control. The role configuration is then used to select the system-level control mode. This shifts the use of roles from individual behavior specialization to team-level stable control orchestration.

## 3 Problem Formulation

We consider a long-horizon cooperative control problem with $N$ agents interacting in a dynamic and partially observable environment. The agents are connected through an interaction structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $i \in \mathcal{V}$ denotes an agent and each edge $(i, j) \in \mathcal{E}$ denotes physical, spatial, or operational coupling between two agents. We model the environment as a partially observable

multi-agent decision process:

$$\mathcal{M} = \langle \mathcal{S}, \{\mathcal{O}^i\}_{i=1}^N, \{\mathcal{A}^i\}_{i=1}^N, P, R, \gamma \rangle \tag{1}$$

where $\mathcal{S}$ is the global state space, $\mathcal{O}^i$ and $\mathcal{A}^i$ are the local observation and action spaces of agent $i$, $P(s_{t+1} \mid s_t, \mathbf{a}_t)$ is the transition function, $R(s_t, \mathbf{a}_t)$ is the shared task reward, and $\gamma \in (0, 1)$ is the discount factor. At time step $t$, each agent receives a local observation $o_t^i \in \mathcal{O}^i$ and selects an action $a_t^i \in \mathcal{A}^i$. The joint action is denoted as $\mathbf{a}_t = (a_t^1, \ldots, a_t^N)$.

Unlike short-horizon cooperative tasks that mainly optimize episodic task completion, long-horizon stable control requires the system to remain within a safe operating region under persistent disturbances and changing local capabilities. Following the standard view in stability analysis, where system stability is commonly characterized by the deviation from a desired equilibrium or safe set Khalil & Grizzle (2002); Agrawal et al. (2022), we describe the system condition using a vector of task-specific stability indicators:

$$\xi_t = h(s_t, \mathbf{a}_t) \tag{2}$$

where $h(\cdot)$ extracts quantities related to system stability. For example, $\xi_t$ can include voltage deviations and violation ratios in power grids, queue spillback and waiting time in traffic networks, or package tilt and angular velocity in collaborative robotic transport.

Given the admissible stability region $\Xi_{\text{safe}}$, we define a task-specific stability loss:

$$L_{\text{stab}}(s_t, \mathbf{a}_t) = \ell(\xi_t; \Xi_{\text{safe}}), \tag{3}$$

where $\ell(\cdot)$ assigns a non-negative penalty according to the distance or violation of $\xi_t$ from the admissible region. In our experiments, $\ell(\cdot)$ is instantiated by domain-specific safety thresholds, such as voltage bounds, queue constraints, or package-stability constraints. The goal is to learn a joint policy $\pi = (\pi^1, \ldots, \pi^N)$ that maximizes task performance while reducing accumulated instability over a long horizon. We formulate the objective as:

$$J_{\text{stab}}(\pi) = \mathbb{E}_\pi \left[ \sum_{t=0}^{T} \gamma^t \left( R(s_t, \mathbf{a}_t) - \beta L_{\text{stab}}(s_t, \mathbf{a}_t) \right) \right] \tag{4}$$

where $\beta$ controls the trade-off between task reward and stability.

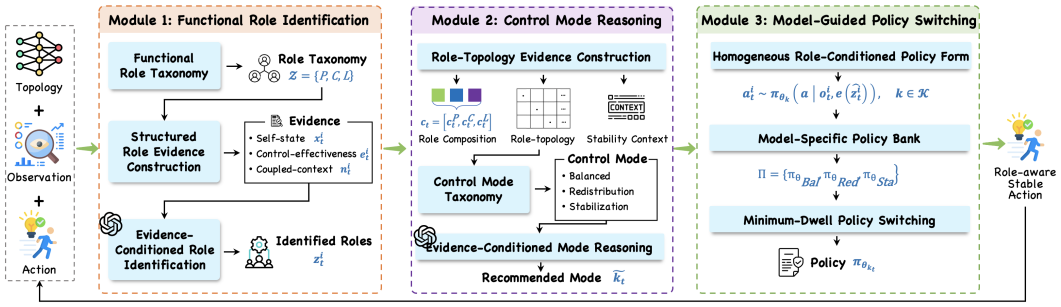


Figure 1: Overview of RoleFit

## 4 Approach

We propose RoleFit, a role-configuration-driven long-horizon stable control method for multi-agent systems. As shown in Figure 1, RoleFit consists of three modules. First, *Functional Role Identification* characterizes each agent according to its current collaborative function. Second, *Control Mode Reasoning* aggregates identified roles with interaction topology and stability context to infer the control mode that matches the current system condition. Third, *Mode-Guided Policy Switching* selects the corresponding homogeneous role-conditioned policy for execution. Through these modules, RoleFit follows the "Horses for Courses" principle: different functional role configurations are handled by control policies with different stability-oriented emphases.This forms a hierarchical architecture with high-level role and mode reasoning and low-level policy execution.

### 4.1 FUNCTIONAL ROLE IDENTIFICATION

This module infers the current functional role of each agent. Instead of using raw observations directly, `RoleFit` converts local observations, action execution information, and coupled system conditions into structured role evidence. The evidence is then used to identify the functional role.

#### 4.1.1 FUNCTIONAL ROLE TAXONOMY

We define 3 functional roles:

$$\mathcal{Z} = \{\mathrm{P}, \mathrm{C}, \mathrm{L}\} \tag{5}$$

where P denotes Primary Contributor, C denotes Compensatory Contributor, and L denotes Limited Contributor. These roles describe what an agent contributes to the team-level control process, rather than whether its local state is simply stable or unstable.

A Primary Contributor provides major task contribution and carries a substantial part of the control responsibility. A Compensatory Contributor mainly reduces system imbalance and compensates the functional deficiency of other agents. A Limited Contributor has constrained effective contribution due to reduced local capability, insufficient regulation margin, execution degradation, or operational restrictions. At time step $t$, the role of agent $i$ is:

$$z_t^i \in \mathcal{Z} \tag{6}$$

The role assignment is dynamic, since the same agent may take different functional roles when its local capability or team function changes.

#### 4.1.2 STRUCTURED ROLE EVIDENCE CONSTRUCTION

We construct role evidence from three observable dimensions: self-state evidence, control-effectiveness evidence, and coupled-context evidence. *(1) Self-state evidence* summarizes recent local observations:

$$x_t^i = \left[o_t^i \| \mathrm{Mean}(o_{t-K+1:t}^i) \| (o_t^i - o_{t-K}^i)\right] \tag{7}$$

where $K$ is the temporal window length and $\|$ denotes concatenation. This term contains the current local observation, the recent average observation, and the local observation trend. *(2) Control-effectiveness evidence* describes whether the control command of an agent is effectively realized by the environment. Let $a_t^i$ be the commanded action produced by the policy, and let $\tilde{a}_t^i$ be the realized action observed after command transmission and actuator execution. The two actions can differ when the command is clipped, saturated, delayed, or disturbed before execution. We define:

$$e_t^i = \left[\|\tilde{a}_t^i\|_2 \| \frac{\|\tilde{a}_t^i\|_2}{\|a_t^i\|_2 + \epsilon} \| \|\tilde{a}_t^i - \tilde{a}_{t-1}^i\|_2\right] \tag{8}$$

where $\epsilon$ is a small constant. The execution ratio indicates whether the agent's intended contribution is fully realized. When the command and the realized action are identical, this ratio becomes one and the evidence reduces to action magnitude and action variation. *(3) Coupled-context evidence* describes the surrounding system condition:

$$n_t^i = \left[\mathrm{Agg}_{j \in \mathcal{N}(i)}(o_t^j) \| \xi_t \| (\xi_t - \xi_{t-K})\right] \tag{9}$$

where $\mathcal{N}(i)$ is the neighbor set of agent $i$, $\mathrm{Agg}(\cdot)$ is a permutation-invariant aggregation function, and $\xi_t$ is the stability indicator defined in Section Problem Formulation. The final role evidence of agent $i$ is:

$$\mathcal{E}_t^i = \left[x_t^i \| e_t^i \| n_t^i\right]. \tag{10}$$

Each field in $\mathcal{E}_t^i$ is normalized and converted into a structured textual representation with fixed field names before role reasoning.

#### 4.1.3 EVIDENCE-CONDITIONED ROLE IDENTIFICATION

Given the structured evidence, `RoleFit` infers the functional role of each agent through an evidence-conditioned role identification:

$$z_t^i = \mathrm{Identification}\left(\mathcal{E}_t^i, \mathcal{D}_{\mathcal{Z}}\right) \tag{11}$$

where $\mathcal{D}_{\mathcal{Z}}$ contains the definitions and decision criteria of the three roles. We implement $\mathrm{Identification}$ with an instruction-following language model and constrained output parsing. The model is required to output exactly one role from $\mathcal{Z}$ based only on the provided evidence. The detailed prompt is provided in the Appendix.

To reduce short-term role oscillation, we smooth the inferred roles over a recent window:

$$\hat{z}_t^i = \mathrm{Majority}\left(z_{t-H+1}^i, \ldots, z_t^i\right) \tag{12}$$

where $H$ is the smoothing window size. The smoothed roles $\{\hat{z}_t^i\}_{i=1}^N$ are used by the next module to construct team-level mode-reasoning evidence.

### 4.2 Control Mode Reasoning

After obtaining the functional roles of agents, `RoleFit` infers the system-level control mode. A single agent role is insufficient for this decision, because the same role count may require different responses depending on the interaction topology and current stability condition. This module therefore builds mode-reasoning evidence from the identified roles, topology, and stability context.

#### 4.2.1 Role-Topology Evidence Construction

We construct the mode reasoning evidence from role composition, role topology, and stability context. The role composition is:

$$\mathbf{c}_t = \left[c_t^{\mathrm{P}}, c_t^{\mathrm{C}}, c_t^{\mathrm{L}}\right], \quad c_t^r = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[\hat{z}_t^i = r], \quad r \in \mathcal{Z} \tag{13}$$

It describes the proportion of Primary, Compensatory, and Limited Contributors in the team.

To capture how roles are distributed over the interaction structure, we construct a role-topology matrix:

$$\mathbf{A}_t(r, r') = \frac{1}{|\mathcal{E}|}\sum_{(i,j)\in\mathcal{E}} \mathbb{I}[\hat{z}_t^i = r, \hat{z}_t^j = r'], \quad r, r' \in \mathcal{Z} \tag{14}$$

This matrix distinguishes isolated role changes from clustered role changes and indicates whether Limited Contributors are coupled with agents that can provide compensation.

We also include the stability context:

$$\mathbf{s}_t = [\xi_t \| (\xi_t - \xi_{t-K})] \tag{15}$$

The final evidence for control mode reasoning is:

$$\Gamma_t = [\mathbf{c}_t \| \mathrm{vec}(\mathbf{A}_t) \| \mathbf{s}_t] \tag{16}$$

#### 4.2.2 Control Mode Taxonomy

We define three system-level control modes:

$$\mathcal{K} = \{\mathrm{Balanced}, \mathrm{Redistribution}, \mathrm{Stabilization}\} \tag{17}$$

*Balanced Control* is used when the team has sufficient effective contributors and the system remains close to the stable region. This mode emphasizes normal task progress while maintaining basic stability. *Redistribution Control* is used when part of the team has limited effective contribution, but the system can still compensate through the remaining agents. This mode emphasizes reallocating control burden from Limited Contributors to Primary and Compensatory Contributors. *Stabilization Control* is used when contribution limitation becomes severe or the stability condition deteriorates. This mode prioritizes safety and stability over short-term task progress. These modes are defined at the system level. All agents operate under the same active control mode, while their role-conditioned policies produce different behaviors according to their individual roles.

#### 4.2.3 EVIDENCE-CONDITIONED MODE REASONING

Given the mode-reasoning evidence $\Gamma_t$, `RoleFit` infers the recommended control mode:

$$\tilde{k}_t = \mathrm{ModeReasoner}\left(\Gamma_t, \mathcal{D}_{\mathcal{K}}\right) \tag{18}$$

where $\mathcal{D}_{\mathcal{K}}$ contains the definitions and decision criteria of the control modes. We implement $\mathrm{ModeReasoner}$ with an instruction-following language model and constrained output parsing. The model is required to output exactly one mode from $\mathcal{K}$ based only on $\Gamma_t$. The detailed prompt is provided in the Appendix. The output $\tilde{k}_t$ is a recommended control mode. The final policy activation and transition process are handled by the policy training and switching module.

### 4.3 MODE-GUIDED POLICY SWITCHING

This module executes the control mode recommended by the mode reasoner. `RoleFit` does not let the language model generate actions. Instead, it selects a mode-specific policy from an offline-prepared policy bank, where each policy corresponds to one control mode and is shared by all agents under that mode.

#### 4.3.1 HOMOGENEOUS ROLE-CONDITIONED POLICY FORM

Each mode-specific policy follows the same homogeneous role-conditioned form. For agent $i$, a policy takes its local observation and role encoding as input:

$$a_t^i \sim \pi_{\theta_k}\left(a \mid o_t^i, e(\hat{z}_t^i)\right), \quad k \in \mathcal{K} \tag{19}$$

where $e(\hat{z}_t^i)$ is the encoding of the inferred role. The policy $\pi_{\theta_k}$ is shared by all agents under mode $k$. Therefore, the specialization is across control modes rather than across agent identities.

#### 4.3.2 MODE-SPECIFIC POLICY BANK

The policy bank is prepared before execution:

$$\Pi = \{\pi_{\theta_k}\}_{k\in\mathcal{K}} = \{\pi_{\theta_{\mathrm{Bal}}}, \pi_{\theta_{\mathrm{Red}}}, \pi_{\theta_{\mathrm{Sta}}}\} \tag{20}$$

Each policy is trained under the role configurations and disturbances associated with its target control mode. For mode $k$, we define the mode-specific reward:

$$\tilde{R}_k(s_t, \mathbf{a}_t) = R_k(s_t, \mathbf{a}_t) - \beta_k L_{\mathrm{stab}}(s_t, \mathbf{a}_t). \tag{21}$$

The corresponding objective is:

$$J_k(\theta_k) = \mathbb{E}_{\pi_{\theta_k}}\left[\sum_{t=0}^{T} \gamma^t \tilde{R}_k(s_t, \mathbf{a}_t)\right]. \tag{22}$$

The Balanced policy keeps a relatively high weight on normal task progress. The Redistribution policy gives more weight to reallocating control burden when Limited Contributors appear. The Stabilization policy gives the highest priority to stability and safety. The concrete reward instantiations and disturbance settings are described in the experimental setup.

#### 4.3.3 MINIMUM-DWELL POLICY SWITCHING

During execution, the mode reasoner outputs a recommended mode $\tilde{k}_t$. To avoid overly frequent mode changes, `RoleFit` updates the active mode only after the current mode has been executed for at least $D_{\min}$ steps:

$$k_t = \begin{cases} \tilde{k}_t, t - t_{\mathrm{last}} \geq D_{\min} \\ k_{t-1}, \mathrm{otherwise} \end{cases} \tag{23}$$

where $t_{\mathrm{last}}$ is the time step of the most recent mode switch. After the active mode is determined, actions are generated by the corresponding policy:

$$\mathbf{a}_t \sim \prod_{i=1}^{N} \pi_{\theta_{k_t}}\left(a^i \mid o_t^i, e(\hat{z}_t^i)\right). \tag{24}$$

This design keeps the switching rule simple and avoids introducing additional action interpolation heuristics.

| Baseline | Power Grid | | Traffic Signal | | Multi-walkers | |
|---|---|---|---|---|---|---|
| | TSR | LSR | TSR | LSR | TSR | LSR |
| MAPPO | 62% | 68% | 57% | 70% | 80% | 73% |
| HAPPO | 58% | 69% | 54% | 64% | 76% | 54% |
| HASAC | 66% | 73% | 53% | 56% | 42% | 57% |
| Heuristic | 63% | 70% | 58% | 66% | 79% | 66% |
| LLMASC | 64% | 72% | 55% | 68% | 82% | 69% |
| **`RoleFit`** | **83%** | **82%** | **93%** | **95%** | **92%** | **88%** |

Table 1: Performance comparison across 3 environments.

# 5 EXPERIMENTS

We evaluate `RoleFit` through baseline comparisons (RQ1), module ablations and LLM-backbone analysis (RQ2), and robustness tests under different disturbance severities and team sizes (RQ3).

We evaluate `RoleFit` on power-grid voltage control Wang et al. (2021b), traffic signal control Krajzewicz et al. (2012); Ault & Sharon (2021), and MultiWalker Gupta et al. (2017); Terry et al. (2021). These tasks involve voltage regulation, intersection coordination, and cooperative package transport, respectively. Disturbances are introduced during execution to evaluate adaptation to changing operating conditions. Environment descriptions and disturbance configurations are provided in Appendix B.

We compare `RoleFit` with MAPPO Yu et al. (2022), HAPPO Zhong et al. (2024), HASAC Liu et al. (2024), a domain-specific heuristic controller, and LLMASC Yu et al. (2026). Results are averaged over three random seeds. Baseline configurations and tuned hyperparameters are provided in Appendix C.

We use two metrics: Task Success Rate and Long-Term Stability Rate. Given $M$ evaluation episodes, let $S_m$ denote whether episode $m$ completes the task, and let $Q_m$ denote whether the stability indicator $\xi_t$ remains within the admissible range throughout the episode. Two metrics are:

$$\text{TSR} = \frac{1}{M}\sum_{m=1}^{M} S_m, \quad \text{LSR} = \frac{1}{M}\sum_{m=1}^{M} Q_m. \tag{25}$$

Higher TSR indicates better task completion, and higher LSR indicates better long-horizon stability. Environment-specific success criteria and operational thresholds are detailed in Appendix E.4.

The default role and mode reasoners use `claude-opus-4-8`. Each of the three mode-specific policies is trained for $2 \times 10^6$ environment steps, giving $6 \times 10^6$ steps in total per policy bank. Architecture, optimization, and reasoning settings are provided in Appendix C.2.

# 6 EXPERIMENTAL RESULTS

## 6.1 RQ1: COMPARISON WITH BASELINES

Table 1 shows that `RoleFit` achieves the highest mean TSR and LSR across the three environments. Compared with the strongest MARL baseline for each metric, TSR improves by 17, 36, and 12 percentage points on power-grid control, traffic signal control, and MultiWalker, respectively; the corresponding LSR gains are 9, 25, and 15 percentage points. `RoleFit` also outperforms LLMASC across all six metrics. These comparisons evaluate the complete systems rather than isolating the effect of role reasoning. Runtime measurements are provided in Appendix D.1.

## 6.2 RQ2: ABLATION STUDY AND LLM ANALYSIS

Table 2 evaluates two module ablations. *w/o Module 1* removes both role inference and role-conditioned policy inputs. *w/o Module 2* retains role information but replaces the LLM mode reasoner with a fixed rule based on role counts and stability thresholds. Both variants reduce mean

| Variants | Power Grid | | Traffic Signal | | Multi-walkers | |
|---|---|---|---|---|---|---|
| | TSR | LSR | TSR | LSR | TSR | LSR |
| w/o Module 1 | 67% | 69% | 43% | 57% | 44% | 76% |
| w/o Module 2 | 71% | 72% | 69% | 77% | 88% | 79% |
| GPT-5.6-sol | 81% | 78% | 92% | 90% | 92% | 89% |
| DeepSeek-V4 | 80% | 77% | 85% | 85% | 95% | 88% |
| Claude–5 | 79% | 78% | 86% | 86% | 93% | 89% |
| **RoleFit** | **83%** | **82%** | **93%** | **95%** | **92%** | **88%** |

Table 2: Ablation study.

| Baseline | Low Severity | | High Severity | |
|---|---|---|---|---|
| | TSR | LSR | TSR | LSR |
| MAPPO | 80% | 73% | 32% | 71% |
| HAPPO | 76% | 54% | 44% | 53% |
| HASAC | 42% | 57% | 23% | 50% |
| Heuristic | 79% | 66% | 46% | 65% |
| LLMASC | 82% | 69% | 33% | 69% |
| **RoleFit** | **92%** | **88%** | **75%** | **84%** |

Table 3: Robustness under different disturbance severities.

TSR and LSR across all environments, supporting the utility of these components within the tested architecture. The first ablation measures the combined effect of role inference and conditioning; the second compares against the particular rule-based selector.

We also replace the default LLM while keeping the policy models and training settings unchanged. Across the tested backbones, changes relative to the default range from a decrease of 10 to an increase of 3 percentage points. No backbone performs best across all metrics.

### 6.3 RQ3: Robustness Analysis

Table 3 reports performance under two disturbance severities. `RoleFit` achieves the highest TSR and LSR in both settings. Its gains over the strongest baseline for each metric are 10 and 15 percentage points under low severity, and 29 and 13 percentage points under high severity, respectively. These results support robustness under the tested disturbances. Additional results for team sizes $N \in \{3, 5, 9\}$ are provided in Appendix D.2.

## 7 Conclusion and Limitation

We proposed `RoleFit`, a role-configuration-driven method for long-horizon stable multi-agent control. `RoleFit` identifies agents' functional roles, reasons about the system-level control mode, and executes the selected mode through homogeneous role-conditioned policies. Experiments on power-grid voltage control, traffic signal control, and collaborative robotic transport show that `RoleFit` improves task success and long-term stability under changing disturbances.

This work uses a compact role and control-mode taxonomy, and the reasoning interval is fixed for each environment. While this design keeps `RoleFit` simple and efficient, more fine-grained domains may benefit from richer role definitions and more flexible reasoning schedules. In the future, we will explore adaptive role taxonomies and dynamic reasoning intervals according to changes in system stability and operating conditions.

### AI USE STATEMENT

We used generative AI tools to assist with language polishing, translation, code development, and manuscript revision, including feedback on the clarity and consistency of method descriptions and reported results. The core method and experimental design were developed by the authors. All AI-assisted edits and generated outputs were reviewed by the authors for accuracy and consistency with the underlying implementation and experimental evidence. The authors take full responsibility for the final manuscript, code, claims, results, and conclusions.

### REPRODUCIBILITY STATEMENT

To support reproducibility, Section 4 describes the components of `RoleFit`, and Section 5 presents the experimental setup, compared methods, and evaluation metrics. Appendices B and C provide additional environment descriptions, baseline hyperparameters, and computational resources. Appendix F provides the role-assignment and policy-selection prompts used in the MAPDN environment. The source code will be submitted as part of the supplementary material to facilitate reproduction of the experiments.

## REFERENCES

Devansh R Agrawal, Hardik Parwana, Ryan K Cosner, Ugo Rosolia, Aaron D Ames, and Dimitra Panagou. A constructive method for designing safe multirate controllers for differentially-flat systems. *IEEE Control Systems Letters*, 6, 2022.

James Ault and Guni Sharon. Reinforcement learning benchmarks for traffic signal control. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

Luigi Berducci, Shuo Yang, Rahul Mangharam, and Radu Grosu. Learning adaptive safety for multi-agent systems. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2859–2865. IEEE, 2024.

Pengcheng Chen, Shichao Liu, Xiaozhe Wang, and Innocent Kamwa. Physics-guided multi-agent deep reinforcement learning for robust active voltage control in electrical distribution systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 71(2):922–933, 2023.

Ming Cheng, Hao Chen, Zhiqing Li, Jia Wang, and Senzhang Wang. Role-aware multi-agent reinforcement learning for coordinated emergency traffic control. *Advances in Neural Information Processing Systems*, 38:139014–139036, 2026.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Zihong Gao, Hongjian Liang, Yuanhui Hao, Lei Hao, and Liangjun Ke. Grdc: A unified graph-driven framework for role discovery and communication in multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29441–29449, 2026.

Harsh Goel, Mohammad Omama, Behdad Chalaki, Vaishnav Tadiparthi, Ehsan Moradi Pari, and Sandeep Chinchali. R3dm: Enabling role discovery and diversity through dynamics models in multi-agent reinforcement learning. *arXiv preprint arXiv:2505.24265*, 2025.

Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theories, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):11216–11235, 2024.

Xudong Guo, Daming Shi, Junjie Yu, and Wenhui Fan. Heterogeneous multi-agent reinforcement learning for zero-shot scalable collaboration. *Neurocomputing*, 648:130716, 2025.

Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *International conference on autonomous agents and multiagent systems*, pp. 66–83. Springer, 2017.

Hassan K Khalil and Jessy W Grizzle. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ, 2002.

Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, Laura Bieker, et al. Recent development and applications of sumo-simulation of urban mobility. *International journal on advances in systems and measurements*, 5(3&4):128–138, 2012.

Hoang M Le, Yisong Yue, Peter Carr, and Patrick Lucey. Coordinated multi-agent imitation learning. In *International Conference on Machine Learning*, pp. 1995–2003. PMLR, 2017.

Huiqun Li, Hanhan Zhou, Yifei Zou, Dongxiao Yu, and Tian Lan. Concaveq: Non-monotonic value function factorization via concave representations in deep multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17461–17468, 2024.

Zhongyu Li, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control. *The International Journal of Robotics Research*, 44(5):840–888, 2025.

Jiarong Liu, Yifan Zhong, Siyi Hu, Haobo Fu, Qiang Fu, Xiaojun Chang, and Yaodong Yang. Maximum entropy heterogeneous-agent reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 57333–57374, 2024.

Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

Jiahu Qin, Qichao Ma, Yang Shi, and Long Wang. Recent advances in consensus of multi-agent systems: A brief survey. *IEEE Transactions on Industrial Electronics*, 64(6):4972–4983, 2016.

Wei Ren and Randal W Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5):655–661, 2005.

Dubravko Sabolić, Igor Ivanković, Zoran Bunčec, and Srđan Skok. The april 28 iberian blackout: Technical insights from official sources available in june 2025. In *2025 IEEE PES Conference on Innovative Smart Grid Technologies-Middle East (ISGT Middle East)*, pp. 1–5. IEEE, 2025.

J K Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, et al. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in neural information processing systems*, 34:15032–15043, 2021.

Jianhao Wang, Zhizhou Ren, Tonghan Liu, Chongjie Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations (ICLR)*, 2021a.

Jianhong Wang, Wangkun Xu, Yunjie Gu, Wenbin Song, and Tim C Green. Multi-agent reinforcement learning for active voltage control on power distribution networks. *Advances in neural information processing systems*, 34:3271–3284, 2021b.

Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Roma: Multi-agent reinforcement learning with emergent roles. *arXiv preprint arXiv:2003.08039*, 2020a.

Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. Rode: Learning roles to decompose multi-agent tasks. *arXiv preprint arXiv:2010.01523*, 2020b.

Woodrow Zhouyuan Wang, Andy Shih, Annie Xie, and Dorsa Sadigh. Influencing towards stable multi-agent interactions. In *Conference on robot learning*, pp. 1132–1143. PMLR, 2022.

Hongtao Wei, Siyu Chang, and Jiaming Zhang. Graph-based topological embedding and deep reinforcement learning for autonomous voltage control in power system. *Sensors*, 25(3):733, 2025.

Aaron Wilson, Alan Fern, and Prasad Tadepalli. Bayesian policy search for multi-agent role discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pp. 624–629, 2010.

Jiaxin Wu, Ziqi Wang, Ji Han, Qionglin Li, Ran Sun, Chenhao Li, Yuehan Cheng, Bokai Zhou, Jiaming Guo, and Bocheng Long. A novel data-driven multi-agent reinforcement learning approach for voltage control under weak grid support. *Sensors*, 25(23):7399, 2025.

Rudai Yan and Yan Xu. Multi-objective and multi-agent deep reinforcement learning for real-time decentralized volt/var control of distribution networks considering pv inverter lifetime. *IEEE Transactions on Power Systems*, 40(2):1558–1569, 2024.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 2022.

Zhuohao Yu, Zhe Liu, Tao Ren, Chenxue Wang, Junjie Wang, and Qing Wang. Many minds, one path: Llm-augmented consensus decision for distributed control in multi-agent collaborative stable scenarios. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28014–28022, 2026.

Qingrui Zhang, Hao Dong, and Wei Pan. Lyapunov-based reinforcement learning for decentralized multi-agent control. In *International Conference on Distributed Artificial Intelligence*, pp. 55–68. Springer, 2020.

Yifan Zhong, Jakub Grudzien Kuba, Xidong Feng, Siyi Hu, Jiaming Ji, and Yaodong Yang. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 25(32): 1–67, 2024.

# A Outline

In this appendix, we give details of the experimental setup in Section B. Section C briefly describes how the algorithms are implemented. Appendix D reports runtime measurements and team-size results. In Section E, we visualize a step-by-step comparison to qualitatively illustrate the advantage of `RoleFit`. In Section F, we introduce the prompts we used.

# B Environmental Settings

To evaluate `RoleFit` on long-horizon stable control tasks, we conduct experiments in three multi-agent control domains that require sustained coordination under continuous disturbances: power-grid voltage regulation, traffic signal control, and multi-robot package transport. These platforms are chosen to cover diverse control characteristics—ranging from discrete switching actions to continuous force coordination—while all providing observable system-level state information necessary for role and mode reasoning.

We note that conventional multi-agent reinforcement learning (MARL) benchmarks (e.g., SMAC, MPE) primarily evaluate coordination on short-horizon tasks with clear terminal conditions. These environments do not expose disturbance information or require sustained stability maintenance, making them unsuitable for evaluating long-horizon stable control under continuous perturbations.

### B.1 MULTI-WALKER

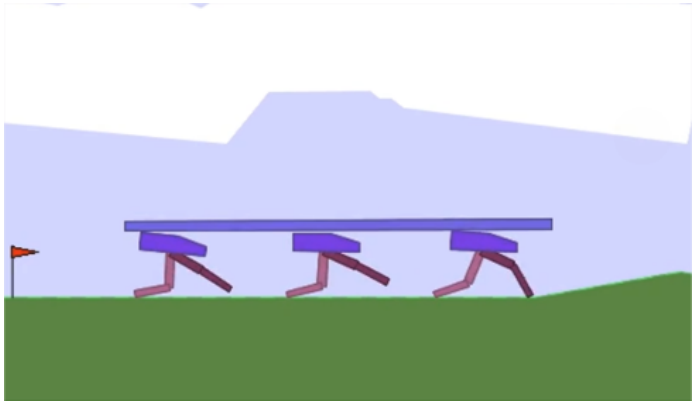

Figure 2: Example of MultiWalker environments

We use the MultiWalker environment Gupta et al. (2017) from the PettingZoo suite Terry et al. (2021) to examine continuous control dynamics, as illustrated in Figure 2. Multiple bipedal agents must coordinate their limb movements to transport a shared package while walking forward without falling. Success requires precise inter-agent coordination to maintain package balance—agents are rewarded for keeping the carrying rod horizontal throughout the episode. This transforms the locomotion task into a stability regulation problem under continuous physical perturbations.

Our MultiWalker evaluation considers both ground-slope disturbances and actuator degradation. Changes in ground slope alter the physical demands of maintaining balance and forward locomotion, while actuator degradation reduces the control authority of selected walkers during specified time windows. These disturbances challenge coordination through changes in terrain conditions and individual agent capability, respectively. The MultiWalker case study in Appendix E examines actuator degradation.

### B.2 SUMO

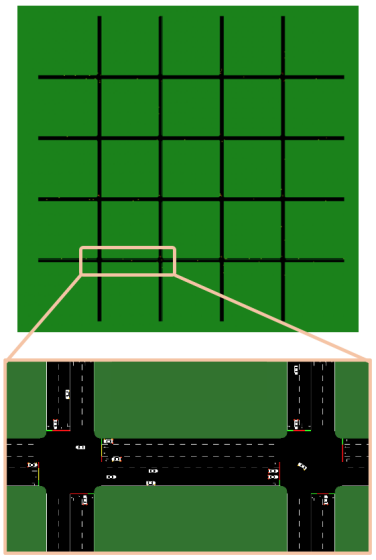

Figure 3: Example of SUMO environments

We use the SUMO (Simulation of Urban MObility) traffic simulator Krajzewicz et al. (2012) with the SUMO-RL interface to evaluate traffic signal control, as shown in Figure 3. Our experiments use standardized network configurations from RESCO Ault & Sharon (2021), a benchmark suite for reinforcement learning in urban traffic management. Each traffic signal acts as an independent agent that observes local traffic states including queue lengths, vehicle counts, and waiting times, aiming to minimize network-wide congestion under time-varying traffic demand.This domain is particularly suitable for evaluating long-horizon stability because traffic networks must maintain smooth flow over extended periods despite continuous fluctuations in demand, incident-induced congestion propagation, and cascading failures where poor control at one intersection can destabilize downstream signals. Disturbances are introduced during execution by increasing traffic demand on selected incoming roads, creating non-uniform traffic pressure across the network. The evaluation examines whether the controllers can maintain traffic flow as the operating conditions change.

### B.3 Multi-Agent Power Distribution Network

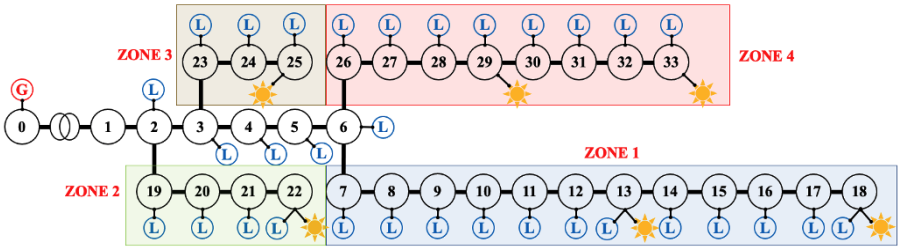


Figure 4: Example of MAPDN environments

We use the Multi-Agent Power Distribution Network (MAPDN) environment Wang et al. (2021b) to evaluate distributed voltage control in electrical power systems, as shown in Figure 4. In this testbed, distributed energy resources (e.g., solar inverters, battery storage) act as independent agents that adjust reactive power to regulate voltage levels and minimize power losses under fluctuating load demand and variable renewable generation. Each agent observes local voltage measurements, power flows, and network topology information. Power grids exemplify long-horizon stability requirements: voltage must remain within tight safety bounds continuously. We introduce disturbances through changes in local load, renewable generation, and the regulation capacity of selected control units. Disturbances occur during the episode rather than only at initialization, requiring the agents to adapt their reactive-power control as operating conditions evolve.

These three domains represent distinct stability challenges: power grids require real-time voltage regulation within tight safety margins; traffic networks must prevent cascading congestion over extended horizons; and multi-robot transport demands continuous balance maintenance under physical perturbations. All three exhibit the key characteristics of long-horizon stable control: prolonged operation under disturbances, safety-critical constraints, and the need for adaptive coordination as system conditions evolve.

## C Experimental Settings

### C.1 Baseline Configurations

We compare `RoleFit` with three MARL methods, a heuristic controller, and an LLM-based control method.

*MAPPO* Yu et al. (2022) uses centralized training with decentralized execution for cooperative multi-agent policy optimization. *HAPPO* Zhong et al. (2024) learns heterogeneous agent policies through sequential policy updates. *HASAC* Liu et al. (2024) uses a heterogeneous-agent soft actor-critic formulation for multi-agent control.

*Heuristic Rules* is a manually designed controller that uses domain knowledge and threshold-based rules to select control actions. *LLMASC* Yu et al. (2026) uses language-model reasoning to support distributed consensus decisions in multi-agent control.

For the baselines, we follow the configurations reported in their original papers where applicable. We search over hyperparameter settings and select the checkpoint with the highest reward. The selected hyperparameters are reported in Table 4. Each method is evaluated using three random seeds, and the reported results are averaged across these runs.

| | **MultiWalker** | **SUMO** | **PowerGrid** |
|---|---|---|---|
| **MAPPO** | | | |
| lr | 0.001 | 0.01 | 0.001 |
| ppo_epoch | 10 | 10 | 5 |
| num_mini_batch | 1 | 4 | 2 |
| **HAPPO** | | | |
| lr | 0.0001 | 0.01 | 0.0001 |
| ppo_epoch | 10 | 4 | 15 |
| num_mini_batch | 2 | 5 | 2 |
| **HASAC** | | | |
| lr | 0.0005 | 0.001 | 0.001 |
| critic_lr | 0.001 | 0.001 | 0.0005 |
| alpha | 0.001 | 0.2 | 0.2 |
| batch_size | 2000 | 1000 | 1000 |
| n_step | 5 | 10 | 10 |
| gamma | 0.99 | 0.95 | 0.99 |

Table 4: Tuned hyper-parameters

### C.2 RoleFit Implementation

**Reasoning configuration.** The role and mode reasoners use `claude-opus-4-8`, with temperature set to 0 and top-$p$ to 1.0. The role-evidence window is $K = 5$. The minimum mode execution length is $D_{\min} = 50$ control steps for traffic signal control and MultiWalker, and $D_{\min} = 20$ for the Multi-Agent Power Distribution Network (MAPDN) environment. The MAPDN role-assignment and policy-selection prompts are provided in Appendix F.

**Policy training.** Each control mode has a separately trained role-conditioned policy shared by all agents under that mode. The three policies use the same backbone architecture, with 128 hidden units per layer. Each policy is trained for $2 \times 10^6$ environment steps, giving a total of $6 \times 10^6$ steps per policy bank. Training uses 5–15 PPO epochs, an actor learning rate of $10^{-3}$, and a critic learning rate of $5 \times 10^{-4}$.

### C.3 Compute Resources

All experiments are conducted on a high-performance computing server featuring dual NVIDIA A6000 GPUs, dual Intel(R) Xeon(R) Platinum 8280 CPU, 503GB RAM, operating under Ubuntu 22.04 LTS.

## D Additional Experimental Results

### D.1 Runtime Analysis

We measure the latency of one `RoleFit` invocation, including role reasoning, mode reasoning, and policy selection. The mean invocation times are 2.07 seconds for power-grid voltage control, 2.09 seconds for traffic signal control, and 2.06 seconds for MultiWalker.

| Method | Metric | $N = 3$ | $N = 5$ | $N = 9$ |
|---|---|---|---|---|
| RoleFit | TSR | 92% | 86% | 72% |
| | LSR | 88% | 74% | 76% |

Table 5: Performance of RoleFit across different team sizes.

The control intervals for power-grid voltage control and traffic signal control are 5 seconds and 180 seconds, respectively. Their mean invocation times are therefore shorter than the corresponding control intervals. These mean timings do not characterize worst-case latency.

### D.2 Team-Size Analysis

We evaluate RoleFit with $N \in \{3, 5, 9\}$ agents. As shown in Table 5, TSR decreases from 92% to 86% and 72% as the team size increases. The corresponding LSR values are 88%, 74%, and 76%. Thus, the method remains operational at the tested larger team sizes, although task success declines. This experiment evaluates RoleFit alone and does not establish superior scalability over competing methods.

## E Case Study

To qualitatively demonstrate the superiority of RoleFit, we provide a detailed step-by-step comparative analysis between baseline MAPPO and our proposed RoleFit approach in a representative uphill locomotion scenario.

### E.1 Case 1: MultiWalker under Degraded Actuation

In the MultiWalker task, three robots must cooperatively transport a package across uneven terrain while preserving both forward progress and formation stability. Our evaluation introduces time-windowed actuator degradation: the control authority of selected walkers is reduced during the episode, with overlapping degradation windows producing a more challenging compound disturbance. This setting captures a realistic cooperative-control failure mode in which individual robots remain operational but can no longer contribute equally to locomotion and load bearing.

The key difficulty is that a local loss of actuation changes the load distribution and motion rhythm of the entire formation. A walker with reduced control authority cannot maintain its nominal gait or speed, and the remaining walkers must adapt their actions to compensate for the weakened teammate. Effective coordination therefore requires the team to adjust its collective pace, redistribute the carrying load, and suppress excessive package oscillation. Otherwise, aggressive forward motion by unaffected walkers can increase package tilt, break formation cohesion, and eventually cause the package to fall.

Conventional policies are prone to maintaining their nominal control patterns after the degradation occurs. Because they do not explicitly interpret the changing contribution of individual walkers, their responses can become poorly synchronized: stronger walkers continue advancing while the weakened walker lags behind, causing large package-angle fluctuations and, in severe cases, premature task failure.

In contrast, RoleFit identifies the evolving cooperative state from the observed motion and stability signals and selects a coordination strategy appropriate for the current disturbance stage. By recognizing which walkers are impaired and how the package and formation are affected, the framework guides the team toward a more conservative and compensatory control pattern. The resulting behavior maintains a more balanced formation, reduces package oscillations, and preserves forward locomotion for a longer duration. These qualitative observations are consistent with the improved task-success and long-term stability results reported in our experiments.

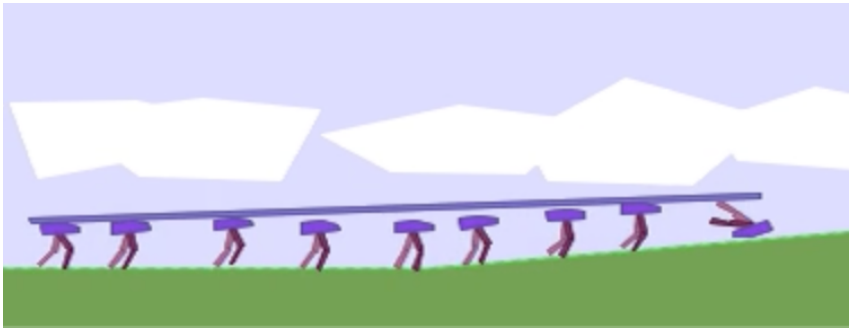

Figure 5: Example of MAPDN environments

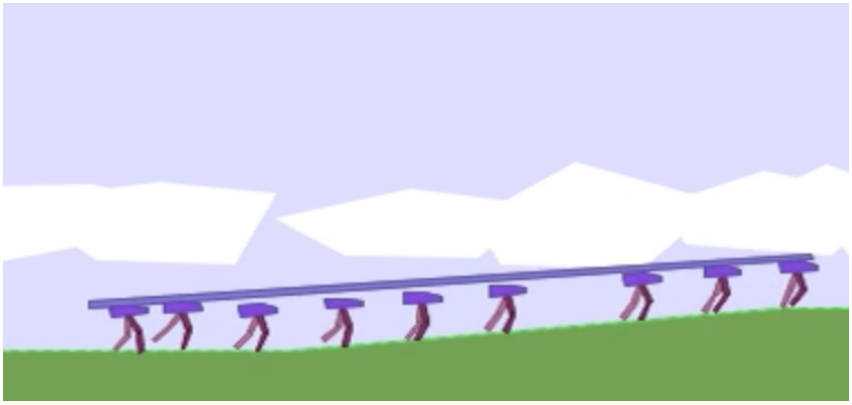

Figure 6: Example of MAPDN environments

### E.2 Case 2: SUMO Traffic Surge with Signal-Control Delay

In the SUMO traffic-signal-control task, agents correspond to connected intersections in a traffic network. We evaluate robustness under a compound disturbance consisting of a sustained traffic-demand increase and delayed traffic-signal actuation. First, additional vehicles are continuously injected into the network, producing non-uniform demand and progressively increasing queue pressure on selected approaches. Subsequently, the actions of selected traffic lights are delayed, such that an intersection may temporarily execute an earlier signal decision rather than its most recently selected phase.

The demand surge requires the network to accommodate elevated inflow, whereas the action delay reduces the responsiveness of part of the signal-control system. Effective control must therefore distinguish between ordinary congestion and delay-induced queue accumulation, coordinate upstream inflow restriction with downstream discharge, and dynamically redistribute green time toward critical traffic movements. A locally reasonable phase decision may become harmful when delayed, making inter-intersection coordination particularly important.

Conventional policies tend to follow a fixed control behavior that was learned under nominal conditions. When traffic demand rises and delayed intersections cannot respond promptly, queues can accumulate rapidly and propagate to neighboring junctions. This may lead to persistent congestion, increased vehicle waiting time, and inefficient use of available network capacity.

By contrast, `RoleFit` interprets the traffic state at the network level and selects control behavior according to the evolving operating stage. It can identify the transition from normal traffic to demand surge and further to the surge-with-delay condition, thereby coordinating the use of appropriate traffic-control strategies across intersections. This enables the system to balance inflow restriction, queue discharge, and network-wide load redistribution, leading to fewer severely delayed vehicles and more stable long-horizon traffic operation.

### E.3 CASE 3: MAPDN EMERGENCY LOAD RISE WITH CONTROL-UNIT DEGRADATION

The emergency load surge scenario represents a critical power system management challenge that demands coordinated reactive power control among multiple distributed energy resources. In this evaluation case, we simulate a sudden and significant increase in electrical load demand across the power distribution network, combined with degraded control capability at selected distributed units. This compound disturbance creates a particularly challenging operating condition: the network requires stronger reactive-power support while some controllers have a reduced ability to provide it, creating voltage instability risks that require immediate coordinated responses to maintain system stability and prevent cascading failures.

The disturbance has both local and system-wide consequences. A load increase can depress voltages at electrically related buses, while reduced regulation capacity limits the ability of the affected controllers to compensate for these deviations. The optimal power system management strategy therefore involves coordinating reactive power adjustments across multiple nodes to address the sudden load increase.Specifically, the remaining distributed control units must recognize the altered network condition, redistribute reactive-power support, and prevent local voltage violations from developing into widespread instability. This requires sophisticated coordination to balance stability requirements with operational constraints, ensuring that the reactive power response is both adequate and optimally distributed across the network.

Baseline power management algorithms fail to recognize the load surge event and its cascading effects on network-wide voltage stability. These baseline policies continue operating under normal load assumptions, producing relatively small reactive power outputs that are insufficient to address the emergency conditions. Such responses are inadequate when the disturbance changes both power demand and the available control resources, leading to persistent voltage deviations or even cascading violations. These algorithms lack the semantic understanding capabilities necessary to adapt to such critical operational scenarios.

In contrast, `RoleFit` uses the observed network state to identify the current operating condition and guide the selection of an appropriate coordinated control strategy. By adapting the collective response to the severity and type of disturbance, `RoleFit` improves voltage recovery and helps maintain stable network operation over extended periods.

### E.4 EVALUATION METRICS AND TASK-SUCCESS CRITERIA

The following criteria define task success and step-level operational constraint satisfaction in each environment. Let $S_m$ denote task success for episode $m$, and let $Q_{m,t}$ indicate whether the operational constraints are satisfied at step $t$.

**Power-grid voltage control.** A control step satisfies the operational constraints ($Q_{m,t} = 1$) when all bus voltages lie within $[0.95, 1.05]$ p.u. An episode is task-successful ($S_m = 1$) if the proportion of steps with at least one voltage violation does not exceed $5\%$.

**Traffic signal control.** A vehicle is classified as congested when its cumulative waiting time exceeds $500$ seconds. Task success requires the peak number of congested vehicles to remain at or below $15\%$ of the concurrent vehicles in the network. A control step satisfies the operational constraint ($Q_{m,t} = 1$) when the system-wide total waiting time is below $\tau_{\text{wait}}$.

**MultiWalker.** An episode is task-successful ($S_m = 1$) if all walkers survive and the package is not dropped before the end of the episode. A control step satisfies the operational constraint ($Q_{m,t} = 1$) when the absolute package tilt is below $5^\circ$.

## F LLM SETTINGS

Following is the prompt we use in MAPDN Environment.The concrete data within the prompt is provided by the runtime environment during execution. Formatted prompt can be found in codes.

### F.1 MAPDN ROLE-ASSIGNMENT PROMPT

You are an analyst for voltage regulation in an Active Distribution Network (MAPDN). The following presents a snapshot of the current status of all distributed control units. The objective is to keep the per-unit voltage values ($v_m$) of all nodes within the compliant range of $[0.95, 1.05]$.

Compare all nodes holistically and assign exactly one role to each node.

**Role definitions**

- `Compensatory Contributor` : The node voltage is compliant with only a modest deviation from the nominal value, andthe node is not currently critical to network-wide stability.
- `Primary Contributor` : The node remains controllable and plays an important stabilizing, regulating, orbuffering role for its local area or neighboring nodes.
- `Limited Contributor` : The node voltage is outside the compliant range, close to a security boundary withclear risk, or is substantially degrading network-wide stability.

**Current network snapshot**

```
Control step: <CURRENT_STEP>

Node <AGENT_ID_1>
  - Control-unit index: <AGENT_INDEX_1>
  - Connected bus: <BUS_ID_1>
  - Voltage: <VM_PU_1> pu
  - Voltage deviation from 1.0 pu: <>
  - Within [0.95, 1.05]: <>
Node <...>
...
```

**Classification guidelines**

1. Check compliance first. A node is non-compliant when $v_m < 0.95$ or $v_m > 1.05$.
2. Classify all non-compliant nodes as `Limited Contributor`. A node close to a limit (e.g., $v_m < 0.96$ or $v_m > 1.04$) may alsobe classified as `Limited Contributor` when its deterioration risk is evident.
3. Classify a compliant but influential node as `Primary Contributor` if it has a noticeable voltage deviation or is important for maintaining local voltage stability.
4. Classify all remaining non-critical compliant nodes as `Compensatory Contributor`.

Return JSON only, with one role for every node:

```
"roles": {
  "<AGENT_ID_1>": "...",
  ...
}
```

### F.2 MAPDN POLICY-SELECTION PROMPT

You are a policy-review expert for voltage control in an Active Distribution Network (MAPDN).

You are given the current role assignment, global voltage statistics, and recent network-level trends. Your task is not to re-assign node roles. Instead, select the most appropriate policy model for the current operating condition.

**Available policy stages**

- `balanced`: Baseline policy for normal operation or very mild disturbances.
- `redistribution`: Moderate policy for a clear but relatively localized voltage risk.
- `stabilization`: Aggressive policy for multiple problem nodes, spreading voltage violations, or rapidly increasing recovery pressure.

**Current control state**

```
Current step: <CURRENT_STEP>
Current confirmed policy stage: <>

Role assignment:
  - <AGENT_ID_1>: ...
  - ...
```

**Operating context**

```
Scenario: <SCENARIO_OR_UNKNOWN>
Current disturbance label:<>
```

**Global voltage summary**

```
Number of controlled nodes: <>
Voltage security range: [0.95, 1.05] pu
Current minimum bus voltage: <MIN_VM_PU>
Current maximum bus voltage: <MAX_VM_PU>
Current mean bus voltage: <MEAN_VM_PU>
Number of lower-bound violations: <>
Number of upper-bound violations: <>
Number of problem nodes: <>
Number of support nodes: <>
Number of routine nodes: <>
```

**Recent network-level trend**

```
Observation window: <>
Minimum voltage: <>
Maximum voltage: <>
Number of voltage violations: <>
```

**Decision guidelines**

1. Consider role distribution, current voltage values, and recent trends jointly; the number of problem nodes alone is insufficient.
2. Select `redistribution` when a localized risk is apparent, including when voltages are trending toward a security bound despite no current violation.
3. Select `stabilization` when violations are spreading, the violation count is increasing, safe coverage is decreasing, or the weakest voltage continues to deteriorate.
4. Select `balanced` only when the network is compliant, stable over the recent observation window, and not trending toward either voltage bound.
5. If some fields are unavailable, treat them as unknown and rely on the available role and voltage information.

Return JSON only:

```
{"target_stage": ...}
```