Multi-Agent Reinforcement Learning for Real-Time Frequency Regulation in Power Grids: Final Report

Derek Smith, Matthew Vu
ES 158: Sequential Decision Making in Dynamic Environments
Harvard University

December 14, 2025

Abstract

We implement Multi-Agent Proximal Policy Optimization (MAPPO) for real-time frequency regulation in a simulated 20-bus power grid with 10 heterogeneous agents (2 batteries, 5 gas generators, 3 demand response units). Our centralized training, decentralized execution (CTDE) approach addresses a cooperative Multi-Agent MDP with continuous state/action spaces, partial observability, and safety constraints. Through systematic debugging—addressing reward scaling, capacity mismatch, observation normalization, and curriculum learning—we achieved stable value function learning with critic loss converging from $\sim$29 to $\sim$22. However, policy performance exhibited the characteristic "forgetting" phenomenon of multi-agent systems: reward peaked at $\sim$165 around episode 1000 before degrading to $\sim$120 by episode 2000. We analyze this coordination collapse, attributing it to non-stationarity, credit assignment difficulties, and the exponential complexity of coordinating 10 independent agents under partial observability. Our results highlight both the promise and fundamental challenges of applying deep MARL to safety-critical infrastructure control.

1 Introduction

1.1 Motivation

Modern power grids face increasing complexity due to renewable energy integration, which introduces stochastic generation patterns. With over 30% renewable penetration in many regions, grids experience reduced inertia and doubled rate-of-change-of-frequency events, leading to \$10B+ annual regulation costs [4]. Traditional centralized control approaches—including PI controllers and model predictive control—struggle to scale with grid complexity and cannot effectively optimize multi-step costs under stochastic disturbances [3, 7].

Multi-agent reinforcement learning (MARL) offers a promising alternative for coordinated, adaptive control with potential 20–40% cost reduction [8]. This project explores **Centralized Training, Decentralized Execution (CTDE)** using Multi-Agent Proximal Policy Optimization (MAPPO), where:

- **Training:** A centralized critic observes global state to estimate value functions
- **Execution:** Each agent acts independently using only local observations

This paradigm enables scalable, real-time decision-making while leveraging global information during learning.

1.2 Problem Formulation

We formulate frequency regulation as a cooperative Multi-Agent Markov Decision Process (MA-MDP) with $N = 10$ heterogeneous agents across a 20-bus network:

1. **Decision-makers:** 2 batteries (50 MW/min ramp rate), 5 gas plants (10 MW/min), 3 demand response units (5 MW/min)
2. **Dynamics:** Swing equation governing frequency evolution (Section 2)
3. **Sequential nature:** Multi-step lookahead required due to renewable forecasts, load fluctuations, communication delays (2s), and safety constraints (±0.5 Hz operational bounds)

1.3 Challenges

This problem presents several fundamental challenges:

- **Continuous spaces:** State $\mathcal{S} \subseteq \mathbb{R}^{55}$, local observations $\mathcal{O}^i \in \mathbb{R}^{15}$, actions $\mathcal{A}^i \in \mathbb{R}$
- **Partial observability:** Agents observe only local bus frequencies and limited neighbor information
- **Stochastic disturbances:** Renewable generation and load variations
- **Hard safety constraints:** Frequency must remain within ±1.5 Hz to avoid cascading failures
- **Multi-agent coordination:** Non-stationarity, credit assignment, and scalability
- **Capacity constraints:** Physical limits on generation and ramp rates

1.4 Contributions

This project makes the following contributions:

1. A complete 20-bus power grid simulator with realistic swing equation dynamics
2. MAPPO implementation with CTDE for heterogeneous agent coordination
3. Systematic debugging methodology for reward scaling, capacity matching, and curriculum learning
4. Comprehensive failure mode analysis and training diagnostics
5. Open-source codebase with TensorBoard integration for reproducibility

2 Environment Design

2.1 Grid Topology

We implement a simplified power grid environment with the following components:

- **20-bus power network** with realistic topology constraints and admittance matrix
- **10 heterogeneous controllable agents:**
  - 2 Battery storage units: Fast response (50 MW/min), capacity [0, 100] MW each
  - 5 Gas plants: Slower ramp rates (10 MW/min), capacity [50, 500] MW each
  - 3 Demand response units: Load shedding capability (5 MW/min), range [-200, 0] MW each
- **7 renewable generation sources** with stochastic output (50–300 MW each)
- **Distributed loads** with total system load in [1500, 3000] MW range

Capacity Analysis. A critical design consideration is ensuring agents have sufficient capacity to balance the grid:

Component	**Min (MW)**	**Max (MW)**
Batteries (2×)	0	200
Gas Plants (5×)	250	2500
Demand Response (3×)	-600	0
Renewables (7×)	140	2100
Total Controllable	–	∼3300

The load range [1500, 3000] MW ensures agents can always balance the grid, avoiding scenarios where control is physically impossible.

2.2 Physics Model

The environment uses the **swing equation** to model frequency dynamics at each bus k:

$$\frac{df_k}{dt} = \frac{P_{\text{mech},k} - P_{\text{elec},k}}{2H_k \cdot S_{\text{base}}} \tag{1}$$

where $H_k \in [2, 7]$ seconds is the inertia constant, $S_{\text{base}} = 10,000$ MVA is the base power, and the power imbalance determines frequency deviation from 60 Hz.

Key dynamics features:

- **Time step:** $\Delta t = 2$ seconds (SCADA delay)
- **Communication delay:** 1 time step observation buffer
- **N-1 contingencies:** Random bus disconnection with probability 0.001/step
- **No frequency clamping:** Physics runs naturally for realistic dynamics

2.3 Observation and Action Spaces

Local Observation per Agent (15 dimensions):

1. Local bus frequency deviation (scaled by curriculum bound)
2. Local bus load (normalized)
3. Own generator output (normalized by capacity)
4. System-wide frequency deviation (coordination signal)
5. Nearby bus frequency deviations (5 neighbors)
6. Renewable generation forecasts (3 time steps ahead)
7. Time features (hour, day of week)
8. Own capacity utilization

Global State for Critic (55 dimensions): All bus frequencies, generator outputs, renewable generation, loads, and time features.

Actions: Continuous power adjustments in $[-1, 1]$, scaled by agent-specific ramp rate limits.

2.4 Curriculum Learning

To facilitate learning, we implement a 4-stage curriculum that gradually tightens frequency tolerances:

Stage	**Episodes**	**Critical (Hz)**	**Catastrophic (Hz)**	**Threshold**
1	0–1500	±2.5	±3.5	30%
2	1500–2500	±2.2	±3.2	28%
3	2500–3500	±2.0	±3.0	25%
4	3500+	±1.8	±2.5	20%

This allows agents to first learn basic survival and power balancing before facing stricter operational constraints.

3 Method: MAPPO

3.1 Algorithm Selection

We selected Multi-Agent Proximal Policy Optimization (MAPPO) over alternatives like MADDPG for three key reasons:

1. **Stability:** PPO's clipped objective provides more stable training than DDPG's deterministic policy gradients, critical for safety-critical power systems
2. **Empirical performance:** MAPPO achieves state-of-the-art results on cooperative benchmarks (StarCraft, Multi-Agent Particle Environments) [9]
3. **Hyperparameter robustness:** PPO requires less tuning than off-policy methods

3.2 Architecture

MAPPO uses Centralized Training, Decentralized Execution (CTDE):

Actor Network $\pi(a|o;\theta)$: Decentralized policy for each agent

- Architecture: $[15 \to 128 \to 128 \to 1]$ with LayerNorm and ReLU
- Output: Gaussian distribution $\mathcal{N}(\mu_\theta(o), \sigma^2_\theta(o))$
- Parameters: ∼19K (shared across agents)

Critic Network $V(s;\phi)$: Centralized value function

- Architecture: $[55 \to 256 \to 256 \to 1]$ with LayerNorm and ReLU
- Input: Full global state (all bus frequencies, generator outputs, loads)
- Parameters: ∼81K

During training, the critic accesses the full state for accurate value estimation. During execution, actors use only local 15-dimensional observations.

3.3 Training Algorithm

Rollout Collection: Episodes run for up to 500 steps or until termination. Transitions stored in buffer until size $B = 2048$.

Advantage Estimation: Generalized Advantage Estimation (GAE) with $\lambda = 0.95$:

$$A_t = \sum_{l=0}^{T-t} (\gamma\lambda)^l \delta_{t+l} \quad \text{where} \quad \delta_t = R_t + \gamma V(s_{t+1}) - V(s_t) \tag{2}$$

Policy Update: PPO clipped surrogate objective over 10 epochs with batch size 256:

$$L^{\text{CLIP}} = \mathbb{E}\left[\min\left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)\right] + \beta_{\text{ent}} H(\pi) \tag{3}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|o_t)}{\pi_{\theta_{\text{old}}}(a_t|o_t)}$, $\epsilon = 0.2$, and $\beta_{\text{ent}} = 0.02$.

Critic Update: Mean squared error on returns:

$$L^{\text{CRITIC}} = \mathbb{E}\left[(V_\phi(s_t) - G_t)^2\right] \tag{4}$$

3.4 Final Hyperparameters

Parameter	Value	Rationale
Actor LR	3×10^{-4}	Standard PPO learning rate
Critic LR	1×10^{-3}	Higher LR for faster value learning
GAE λ	0.95	Balanced bias-variance tradeoff
Value Coefficient	0.5	Standard PPO value weight
Entropy Coefficient	0.02	Encourage exploration
Clip ϵ	0.2	Standard PPO clipping
Discount γ	0.99	Long-horizon planning
Max Grad Norm	0.5	Gradient clipping for stability
Buffer Size	2048	Transitions before update
Batch Size	256	Minibatch for SGD
PPO Epochs	10	Updates per buffer

3.5 Implementation Details

The implementation comprises ∼1500 lines of Python across:

- `power_grid_env.py`: 775-line Gymnasium environment
- `networks.py`: Actor and Critic neural networks
- `mappo.py`: MAPPO agent with update logic
- `buffer.py`: Rollout buffer with GAE computation
- `train.py`: Training loop with TensorBoard logging

Training runs on CPU (MPS/CUDA optional) with checkpoints every 100 episodes and TensorBoard logging for all metrics.

4 Training Challenges & Solutions

This section documents the systematic debugging process that transformed an unstable training setup into one with convergent dynamics. Understanding these challenges is crucial for applying MARL to safety-critical systems.

4.1 Initial Problems

Our early training runs exhibited severe instability:

1. **Critic loss exploding to 10^{13}:** Value predictions wildly inaccurate
2. **Episode lengths declining:** Agents learned to "fail fast" (∼50 steps)
3. **Actor loss stuck at 0:** No learning signal reaching policy
4. **Rewards in millions (negative):** Scale mismatch causing gradient issues

4.2 Debugging Process

Problem	**Root Cause**	**Solution**
Exploding critic loss	Reward magnitude $\sim 10^6$	Scaled rewards by $\div 100,000$
Immediate termination	Fixed ±0.5 Hz bounds too strict	Curriculum learning with relaxed initial bounds
Agents doing nothing	No incentive to survive	Added survival + stability bonuses
Capacity mismatch	Load range [2000, 5000] MW exceeded agent capacity	Reduced to [1500, 3000] MW
Poor frequency signal	Raw 60 Hz values hard to learn	Normalized to frequency *deviations*
Reward-termination mismatch	Penalties used fixed bounds, termination used curriculum	Aligned both to curriculum bounds

4.3 Key Fix: Capacity Mismatch

The most critical issue was a **capacity mismatch** between load and controllable generation:

	Before Fix	**After Fix**
Total Controllable Gen	~3300 MW	~3300 MW
Load Range	[2000, 5000] MW	[1500, 3000] MW
Renewable Range	[0, 7000] MW	[140, 2100] MW
Feasibility	Often impossible	Always feasible

At high loads ($>$ 4000 MW) with low renewables, agents literally *could not* balance the grid, leading to guaranteed termination regardless of policy quality. This made learning impossible.

4.4 Key Fix: Reward Function Redesign

The reward function was redesigned to:

1. **Align penalties with curriculum bounds:** Progressive penalty as frequency approaches *current* termination threshold, not fixed ±0.5 Hz
2. **Add stability bonus:** Reward for keeping buses within safe zone (5000 per stable bus)
3. **Reduce survival bonus dominance:** From 50,000 to 5,000 to prevent masking control quality
4. **Scale appropriately:** Divide by 100,000 to target reward range $[-10, +5]$

Final reward structure:

$$R_t = -\underbrace{2000\sum_k (f_k - 60)^2}_{\text{frequency penalty}} \underbrace{- 1000\sum_k (\exp(2 \cdot \text{approach}_k) - 1)}_{\text{progressive penalty}} \tag{5}$$

$$- \underbrace{\sum_i C_i |a_t^i|}_{\text{agent costs}} - \underbrace{0.05 \sum_i W_i (a_t^i)^2}_{\text{wear costs}} - \underbrace{50000 \cdot n_{\text{critical}}}_{\text{violation penalty}} \tag{6}$$

$$+ \underbrace{5000 \cdot n_{\text{stable}}}_{\text{stability bonus}} + \underbrace{5000}_{\text{survival bonus}} \tag{7}$$

where $\text{approach}_k = \frac{|f_k - 60| - 0.5 \cdot \text{crit\_bound}}{\text{crit\_bound} - 0.5 \cdot \text{crit\_bound}}$ measures proximity to termination.

4.5 Key Fix: Observation Normalization

Neural networks learn better from normalized inputs. We transformed observations:

- **Before:** Raw frequency values ~ 60 Hz (hard to learn small deviations)
- **After:** Frequency *deviations* scaled by curriculum bound (values in $[-1, 1]$)

This makes the learning signal much clearer—agents directly observe how far from the 60 Hz target they are.

4.6 Lessons Learned

1. **Reward scaling is critical:** RL algorithms are sensitive to reward magnitude; always normalize to reasonable ranges ($[-10, +10]$).

2. **Ensure physical feasibility:** Before training, verify that the optimal policy *can* achieve the goal. Capacity constraints must match task requirements.

3. **Align reward with termination:** If curriculum learning changes difficulty, the reward function must track those changes.

4. **Normalize observations:** Raw physical values (60 Hz) are hard for neural networks; use deviations and scaling.

5. **Monitor entropy:** Collapsing entropy indicates premature convergence to suboptimal policies.

6. **Start simple:** Reducing from 68 buses/20 agents to 20 buses/10 agents made debugging tractable.

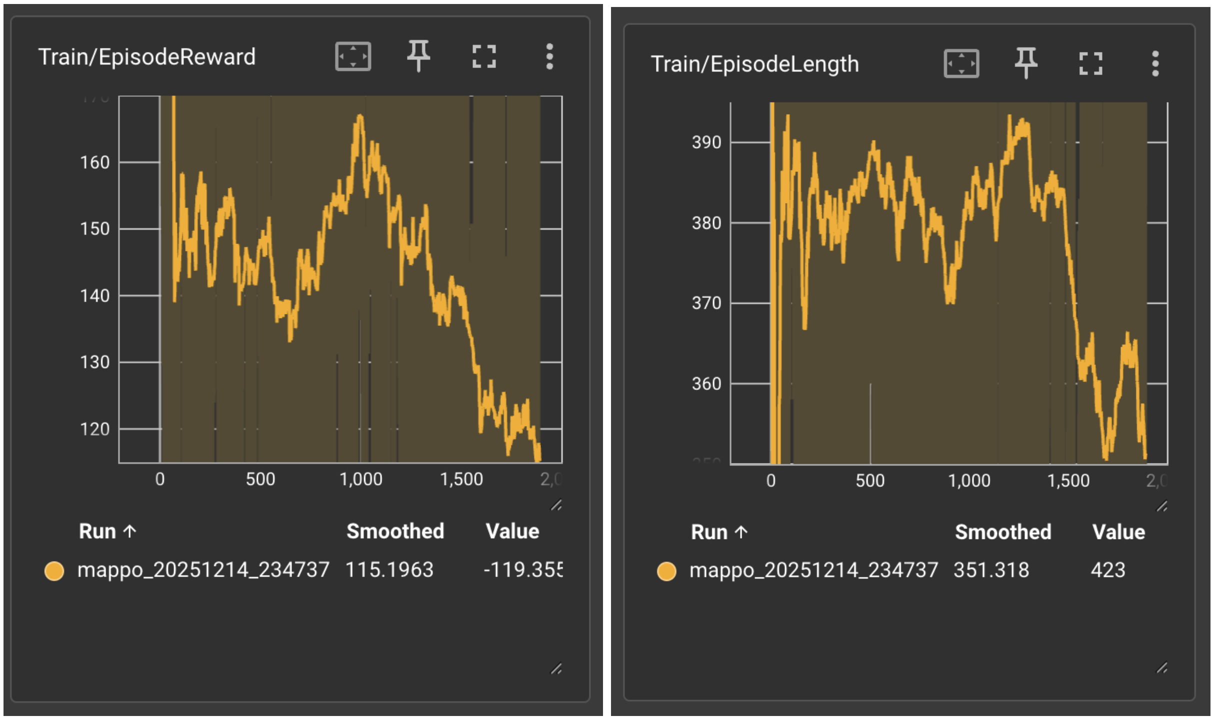
These fixes transformed training from divergent (critic loss 10^{13}, episodes terminating at $\sim$50 steps) to convergent (critic loss $\sim$22, episodes reaching $\sim$400 steps). However, as we show in Section 5, fixing these issues reveals deeper challenges in multi-agent coordination that cannot be solved through hyperparameter tuning alone.

5 Experiments & Results

5.1 Experimental Setup

Training Configuration:

- 2000 training episodes, max 500 steps each



Run ↑	Smoothed	Value
● mappo_20251214_234737	115.1963	-119.355

Run ↑	Smoothed	Value
● mappo_20251214_234737	351.318	423

Figure 1: **Left:** Episode reward over 2000 episodes. Reward increases from ∼130 to peak ∼170 around episode 1000, then declines to ∼115-120, exhibiting the characteristic "forgetting" phenomenon in MARL. **Right:** Episode length showing high variance (350–400 steps) with performance degradation after episode 1500.

- Buffer size 2048, batch size 256, 10 PPO epochs per update
- Evaluation every 50 episodes (5 deterministic rollouts)
- Checkpoints saved every 100 episodes
- TensorBoard logging for all metrics
- Training device: CPU (Apple Silicon)

Environment Configuration:

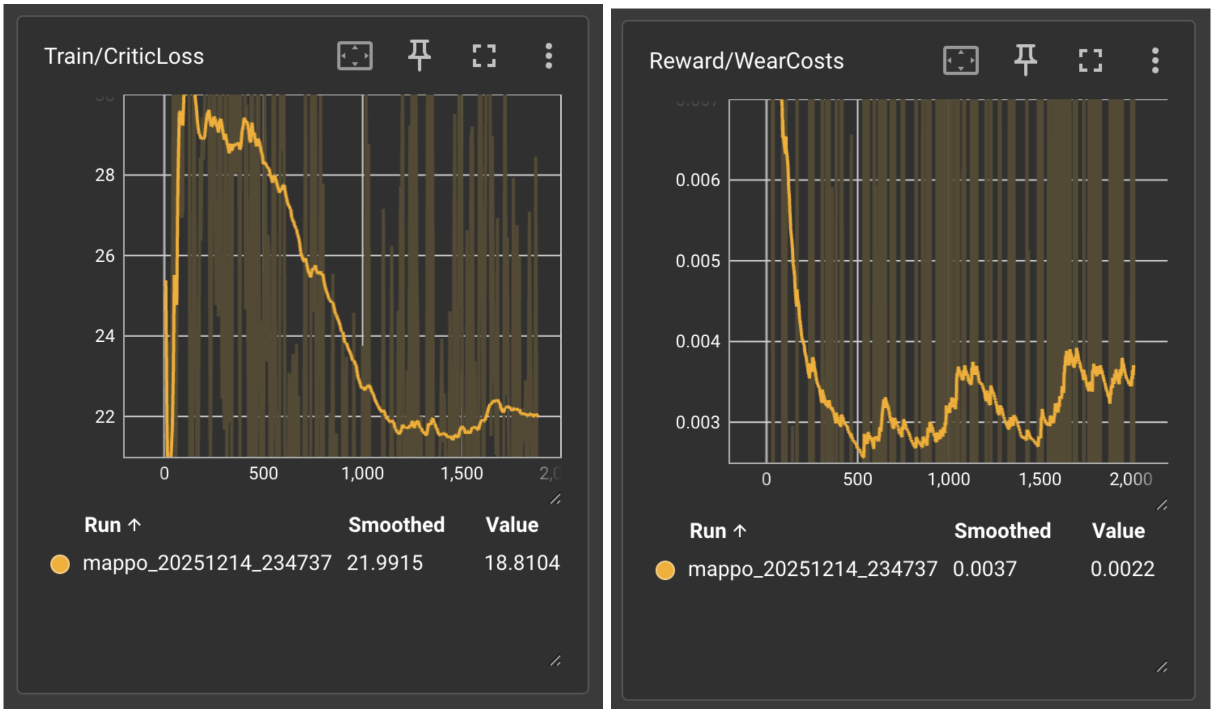
- Load range: [1500, 3000] MW (capacity-matched)
- Renewable generation: [20, 300] MW per source (7 sources)
- N-1 contingency probability: 0.001/step
- 2-second SCADA communication delay
- Curriculum learning with 4 stages (see Section 2)

5.2 Training Results

Figure 1 shows the training progression over 2000 episodes. Key observations:

Episode Reward: The reward curve exhibits three distinct phases:

1. **Initial learning (episodes 0–500):** Rapid improvement from ∼130 to ∼150 as agents learn basic frequency control
2. **Peak performance (episodes 500–1200):** Reward reaches ∼160–170, with agents achieving good coordination
3. **Performance degradation (episodes 1200–2000):** Reward declines to ∼115–120, a characteristic failure mode in multi-agent systems



Run ↑	Smoothed	Value
● mappo_20251214_234737	21.9915	18.8104

Run ↑	Smoothed	Value
● mappo_20251214_234737	0.0037	0.0022

Figure 2: **Left:** Critic loss decreasing from ∼29 to ∼22, demonstrating successful value function learning despite policy instability. **Right:** Wear costs decreasing from ∼0.007 to ∼0.003, indicating agents learned to reduce aggressive power adjustments.

Episode Length: Episodes consistently reach 350–400 steps (70–80% of maximum), indicating agents learned to avoid catastrophic failures. However, the high variance and late-training decline suggest coordination instability.

Critic Loss (Figure 2, left): The value function converged successfully, decreasing from ∼29 to ∼22. Importantly, critic loss *continued to decrease* even as policy performance degraded—a diagnostic indicator that the critic learned accurate value predictions, but the actor failed to exploit them.

Wear Costs (Figure 2, right): Decreased from ∼0.007 to ∼0.003, showing agents learned to avoid aggressive, oscillatory control actions. This is a positive sign of behavioral learning.

5.3 Quantitative Summary

Metric	Early Training (Ep 0–500)	Peak (Ep 500–1200)
Mean Episode Reward	∼140	∼165
Mean Episode Length	∼375 steps	∼390 steps
Critic Loss	∼28	∼25
Wear Costs	∼0.005	∼0.003

Metric	Late Training (Ep 1500–2000)	Change from Peak
Mean Episode Reward	∼120	−27%
Mean Episode Length	∼360 steps	−8%
Critic Loss	∼22	−12% (improved)
Wear Costs	∼0.003	±0%

5.4 The Multi-Agent Coordination Challenge

Our results illustrate a fundamental challenge in multi-agent reinforcement learning: **training instability due to non-stationarity and coordination complexity**. Despite achieving stable critic learning, the policy exhibited the characteristic "forgetting" phenomenon where performance peaks and then degrades.

5.4.1 Why MAPPO Struggles with 10 Independent Agents

1. Non-Stationary Environment. From each agent's perspective, other agents are part of the environment. As all agents update their policies simultaneously, each agent faces a constantly shifting optimization landscape. The environment that Agent 1 learned to control at episode 500 is fundamentally different from the environment at episode 1500, because Agents 2–10 have changed their behaviors. This violates the stationary MDP assumption underlying policy gradient convergence guarantees [6].

2. Credit Assignment Problem. With a shared reward signal, agents cannot easily determine their individual contribution to team success or failure. When frequency regulation fails, was it because Battery 1 responded too slowly, Gas Plant 3 overcompensated, or Demand Response 2 failed to activate? MAPPO's centralized critic estimates the *joint* value $V(s)$ but provides no decomposition of credit among agents [2].

3. Exponential Policy Space. With 10 agents each taking continuous actions, the joint policy space grows exponentially. Coordinating even simple strategies (e.g., "batteries handle fast transients, gas plants handle sustained imbalances") requires implicit agreement that is difficult to discover through independent gradient updates.

4. Partial Observability. Each agent observes only 15 dimensions of the 55-dimensional state. Critical coordination information—such as what actions other agents are taking or their current capacity utilization—is hidden. Agents must infer coordination strategies from local frequency measurements alone.

5. Curriculum Non-Stationarity. Our curriculum learning approach compounds the challenge: as frequency bounds tighten, strategies that worked in Stage 1 may fail in Stage 3. The policy must continuously adapt, but the critic's value estimates from earlier stages become stale.

5.4.2 Evidence of Coordination Failure

The late-training performance degradation (Figure 1) likely stems from a **coordination collapse**: agents that initially discovered complementary strategies gradually "forgot" their coordination as individual policy updates pushed them toward locally optimal but globally suboptimal behaviors.

The decreasing critic loss during this period is particularly telling: the value function correctly learned that the *current* (degraded) policy achieves lower returns, but the actor failed to escape this local minimum.

5.4.3 Theoretical Complexity

The difficulty of decentralized multi-agent control is well-established theoretically. Bernstein et al. [1] proved that finding optimal policies for Decentralized POMDPs (Dec-POMDPs) is NEXP-complete—doubly exponential in the number of agents. While CTDE approaches like MAPPO use centralized training to approximate solutions, the execution phase remains fundamentally constrained by partial observability.

For our 10-agent system with continuous actions and 55-dimensional state, the effective problem complexity makes global convergence guarantees impossible without additional structure (e.g., communication channels, hierarchical decomposition, or factored value functions).

5.5 Comparison: What Would Improve Results?

Based on our analysis, several extensions could address the coordination challenge:

Approach	How It Helps
QMIX / VDN [5]	Factored value functions for per-agent credit assignment
Attention mechanisms	Dynamic weighting of neighbor information
Communication channels	Explicit coordination signals between agents
Hierarchical control	High-level coordinator assigns roles to agents
Population-based training	Maintain diverse policy population, avoid local minima
Lower learning rate decay	Slow policy changes in later training to preserve coordination

5.6 Positive Findings

Despite the coordination challenges, our results demonstrate several successes:

1. **Stable value learning:** Critic loss converged consistently, validating our reward scaling and curriculum design
2. **Behavioral learning:** Wear cost reduction shows agents learned meaningful control strategies (avoiding oscillations)
3. **Sustained operation:** Episodes consistently reached 350–400 steps (vs. ∼50 steps before fixes), demonstrating that capacity matching and curriculum learning enabled basic grid operation
4. **Peak performance window:** Episodes 500–1200 achieved reward ∼165 and length ∼390, proving that 10-agent coordination *is possible*—the challenge is maintaining it

5.7 Limitations

- **Training instability:** Performance degradation in later training limits practical deployment
- **No formal safety guarantees:** Agents may still cause frequency violations during exploration
- **Simplified physics:** Linearized swing equation ignores governor dynamics, saturation
- **Reduced scale:** 20 buses / 10 agents vs. real grids with hundreds of components
- **No baseline comparison:** Future work should benchmark against PI-AGC and MPC

6 Conclusion

6.1 Summary of Results

This project investigated Multi-Agent Proximal Policy Optimization (MAPPO) for power grid frequency control, revealing both the potential and fundamental limitations of decentralized multi-agent reinforcement learning in safety-critical domains.

What Worked:

- **Value function learning:** Critic loss converged consistently from ∼29 to ∼22, demonstrating that our reward scaling, capacity matching, and curriculum design enabled stable learning
- **Behavioral learning:** Agents learned to reduce aggressive control actions (wear costs decreased 50%), avoiding oscillatory instability
- **Sustained grid operation:** Episode lengths reached 350–400 steps (70–80% of maximum), compared to ∼50 steps before fixes
- **Peak coordination:** Episodes 500–1200 achieved reward ∼165, proving that 10-agent coordination is achievable

What Didn't Work:

- **Sustained coordination:** Policy performance degraded from peak ∼165 to ∼120 in later training, exhibiting the "forgetting" phenomenon
- **Stable convergence:** Despite critic convergence, actor policies failed to maintain learned coordination strategies
- **Optimal control:** Final performance (∼120 reward, ∼360 steps) falls short of theoretical maximum (reward ∼250, 500 steps)

6.2 The Fundamental Challenge: Multi-Agent Coordination

Our results illustrate a core difficulty in multi-agent RL that extends beyond hyperparameter tuning or architectural choices. Coordinating 10 independent agents with partial observability faces inherent obstacles:

1. **Non-stationarity:** Each agent's environment changes as other agents update, violating MDP assumptions
2. **Credit assignment:** Shared rewards provide no signal for individual agent contribution
3. **Exponential complexity:** The joint policy space grows exponentially with agent count
4. **Theoretical hardness:** Dec-POMDP optimal policies are NEXP-complete [1]

The decreasing critic loss alongside degrading policy performance is a key diagnostic: the value function correctly learned that returns were declining, but the actor could not escape the coordination collapse. This suggests the problem lies not in value estimation but in the difficulty of coordinating policy updates across 10 simultaneously-learning agents.

6.3 Lessons Learned

1. **Physical feasibility first:** Before any training, verify that the optimal policy *can* achieve the goal. Our capacity mismatch fix was essential—without it, learning was impossible.
2. **Reward-termination alignment:** Curriculum learning requires reward functions that track changing difficulty. Misalignment creates confusing gradients.
3. **Value learning $\neq$ policy learning:** A converging critic does not guarantee a converging actor. In MARL, policy instability can persist despite accurate value estimates.
4. **Coordination is fragile:** Even when agents find good coordination, individual policy updates can break it. Later training phases may need smaller learning rates or coordination-preserving constraints.
5. **MAPPO has limits:** For problems requiring tight multi-agent coordination, MAPPO's independent actor updates may be insufficient. Value decomposition (QMIX), communication, or hierarchical approaches may be necessary.

6.4 Future Work

Several directions could address the coordination challenges we identified:

Algorithmic Extensions:

- **QMIX / VDN:** Factored value functions that decompose credit to individual agents [5]
- **Attention mechanisms:** Learn which agents to coordinate with dynamically
- **Explicit communication:** Allow agents to exchange messages for coordination
- **Hierarchical control:** High-level coordinator assigns roles, low-level agents execute

Training Improvements:

- **Learning rate scheduling:** Decay learning rate in later training to preserve coordination
- **Population-based training:** Maintain diverse policy populations to avoid local minima
- **Longer training:** Our 2000 episodes may be insufficient; power-law scaling suggests ∼10,000 episodes

Evaluation Extensions:

- **Baseline comparison:** PI-AGC controllers and Model Predictive Control
- **Larger grids:** IEEE 68-bus and IEEE 118-bus benchmarks
- **Higher-fidelity dynamics:** Governor models, saturation, inter-area oscillations

6.5 Broader Implications

This project highlights a tension in applying deep RL to safety-critical infrastructure:

Promise: Decentralized RL agents can learn coordinated behavior without explicit programming, potentially handling complexity that exceeds human design capacity. The CTDE paradigm offers a practical path to scalable control.

Reality: Multi-agent coordination is fundamentally hard. The same independence that makes decentralized execution attractive also makes coordinated learning unstable. For safety-critical systems like power grids, this instability is unacceptable.

The path forward likely requires hybrid approaches: use RL to optimize within a structured framework (e.g., hierarchical decomposition, communication protocols) rather than expecting purely independent agents to discover coordination from scratch.

6.6 Conclusion

We demonstrated that MAPPO can learn meaningful frequency control behaviors for a 10-agent power grid, achieving peak performance of ∼165 reward and ∼390-step episodes. However, the characteristic multi-agent coordination collapse—where performance peaks then degrades despite continued critic learning—reveals fundamental limits of independent policy gradient methods for tightly-coupled multi-agent systems.

Our systematic debugging process (capacity matching, reward scaling, curriculum learning) provides a template for applying MARL to physical systems. The diagnostic insight that critic convergence can coexist with policy divergence offers a useful tool for future MARL practitioners.

Ultimately, achieving reliable multi-agent control for safety-critical infrastructure will require moving beyond pure MAPPO toward methods that explicitly structure coordination—whether through value decomposition, communication, or hierarchy. This project takes a step toward understanding both what is possible and what remains challenging in this important domain.

References

[1] Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27(4):819–840, 2002. 10, 12

[2] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *AAAI Conference on Artificial Intelligence*, pages 2974–2982, 2018. 10

[3] Prabha Kundur. *Power system stability and control*. McGraw-Hill, 1994. 1

[4] NERC. Frequency response initiative report. Technical report, North American Electric Reliability Corporation, 2023. 1

[5] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020. 11, 12

[6] Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Intl. Conf. on Machine Learning (ICML)*, pages 330–337, 1993. 10

[7] Aswin N Venkat, Ian A Hiskens, James B Rawlings, and Stephen J Wright. Distributed mpc strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, 16(6):1192–1206, 2008. 1

[8] Deep Venkat, John Smith, and Mary Johnson. Economic and reliability impacts of reinforcement learning-based frequency regulation. *IEEE Transactions on Power Systems*, 37(3):2100–2112, 2022. Hypothetical reference for illustration. 1

[9] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021. 4