Power Grid: A Complex Multi-Player Board Game Environment for Reinforcement Learning

Stephen Hornish
Johns Hopkins University
Baltimore, MD, USA
shornis1@jh.edu

Alhassan S. Yasin
Johns Hopkins University
Baltimore, MD, USA
ayasin1@jhu.edu

***Abstract*—Board games have long served as testbeds for artificial intelligence research, from early tree-search approaches in Chess and Go to modern systems like AlphaGo. However, contemporary strategy board games present distinct challenges for reinforcement learning: multi-agent dynamics, phase-based mechanics, large combinatorial action spaces, and complex strategic dependencies that unfold over long time horizons. Power Grid exemplifies these challenges, requiring players to simultaneously manage economic optimization, spatial network expansion, shared resource markets, and power plant auctions across multiple game phases. This paper introduces a fully observable multi-player RL environment for Power Grid, implemented in the Tabletop Games Framework (TAG) and accessible via PyTAG. We formalize the game as an MDP, design a structured observation space supporting variable player counts (3–6), implement phase-aware action masking, and introduce reward shaping for the game's multi-phase structure. Baseline evaluations of PPO agents against Random and MCTS opponents reveal how performance, episode length, and emergent strategies vary with player count and opponent strength. Our results characterize the environment's difficulty and provide a foundation for future research in complex, phase-based multi-agent games.**

***Index Terms*—Board Games, Reinforcement learning, action masking, Power Grid, proximal policy optimization.**

I. Introduction

Board games have long served as benchmarks for artificial intelligence, offering controlled environments for evaluating search and learning algorithms. Classical domains such as Chess, Go, and Checkers have driven progress in heuristic search, Monte Carlo Tree Search (MCTS), and deep reinforcement learning (RL), culminating in systems that exceed human performance [1]. More recent work has explored modern games—for example, Pandemic via Statistical Forward Planning (SFP) [2] and Ticket to Ride via deep RL [3]. However, many contemporary strategy games remain largely unstudied within RL, despite combining economic management, spatial planning, shared resource markets, and multi-phase decision processes that induce high-dimensional states and large combinatorial action spaces. These games additionally function as abstractions of real-world strategic systems, making them relevant not only for entertainment, but also for applications in decision support, simulation, and war-gaming.

The Tabletop Games framework (TAG) [4] framework and its Python interface PyTAG [5] enable such games to be embedded in standardized gym-style environments with support for both SFP agents and learning-based methods [5]. While TAG covers a diverse set of titles, existing RL studies have primarily focused on short-horizon games or those with relatively simple action structures, such as Sushi Go [6], Exploding Kittens [7], and Tic Tac Toe. In contrast, commercially popular strategy games involve long episodes, multiple interacting subsystems, and phase-dependent action sets that change dynamically throughout gameplay. Power Grid exemplifies this complexity. Players compete to build electrical networks across 3–6 player games, balancing plant auctions, shared resource markets, map-based spatial expansion, and power generation across five distinct game phases. The environment is fully observable but features interdependent decision layers that yield approximately 10^{44} possible states, a structured observation space of variable dimension, and a phase-dependent action space of 178 actions. Victory requires long-term economic planning, spatial reasoning about network connectivity and opponent positioning, and tactical decisions in competitive auctions—making Power Grid a challenging multi-agent RL benchmark that complements existing testbeds.

In this work we introduce a full TAG implementation of Power Grid, exposed through PyTAG and designed for RL evaluation. Our contributions are threefold:

1) We provide a complete TAG implementation of *Power Grid*, including an MDP formulation, a structured observation space supporting 3–6 players, and a phase-aware action space of 178 actions represented via conditional action trees and action masks.
2) We present baseline results for PPO agents trained against Random and MCTS opponents, demonstrating how opponent strength and player count affect win rates, episode length, and cross-player-count generalization.
3) We analyze emergent strategies, including region-selection patterns, resource and plant-type preferences, and the relationship between auction initiation and plant usage, revealing convergent strategic behaviors across learning and planning agents.

These contributions provide the research community with a challenging benchmark for multi-agent RL in phase-based strategy games, together with baseline characterization and initial insights into learned behaviors. The Power

Grid TAG implementation is available at https://github.com/StephenHornish/TabletopGames/tree/Modified-RL. The corresponding PyTAG repository, which includes full documentation and instructions for running the Power Grid environment and reproducing the experiments, is available at https://github.com/StephenHornish/PyTAG.

II. Background

A. Related Work

Research on AI for tabletop games has historically focused on classical domains such as Chess, Go, and Poker, with substantially less attention given to the complex, multi-phase designs characteristic of modern hobby games. The Tabletop Games (TAG) framework addresses this gap by providing a Java-based platform for implementing contemporary tabletop games, analyzing their structural properties, and benchmarking statistical forward planning (SFP) agents [4]. TAG offers a unified API for game states and forward models, reusable components for actions, rules, turn orders, and phases, and a suite of built-in agents including Random, One-Step Look Ahead (OSLA), Rolling Horizon Evolutionary Algorithms (RHEA), and Monte Carlo Tree Search (MCTS). Empirical results across several implemented games (e.g., Love Letter, Uno, Exploding Kittens, Pandemic) show MCTS to be the strongest baseline, while also revealing core challenges for AI in tabletop domains: large and variable action spaces, partial observability, stochasticity, and multi-player interactions. However, the TAG framework is oriented toward planning-based control rather than reinforcement learning (RL), and does not directly provide fixed observation or action spaces suitable for model-free RL training.

Building on TAG, the PyTAG interface extends the framework to Python and introduces gym-style wrappers that support integration with modern RL libraries and vectorized training pipelines [5]. PyTAG provides tools for constructing fixed-format observations, generating structured action trees, and applying action masks to handle the dynamically changing action spaces common in tabletop games. Experiments with PPO across a range of games (Diamant [8], Love Letter [9], Exploding Kittens [7], Stratego [10]) demonstrate the feasibility of applying deep RL methods in these environments but also highlight pervasive difficulties: high branching factors, delayed rewards, long horizons, and degradation in performance as the number of players increases [5]. These findings suggest that modern tabletop games pose substantial challenges for RL.

Exploring Reinforcement learning performance in a modern board game sharing some similarities to Power Grid such as variable player count and map based expansion, Yang et al. [3] studied deep RL in the board game, *Ticket to Ride*, modeling it as a partially observable MDP with stochastic card draws and a high branching factor. Their PPO agents, trained with domain-knowledge features but without heuristic guidance, outperformed several handcrafted baselines and achieved strong self-play performance. While their work focuses on partial observability and stochasticity, the present study targets a deterministic, fully observable environment with complex phase mechanics and dynamically changing action masks. Together, these prior works illustrate both the promise and the difficulty of applying RL to modern tabletop games and motivate the development of a Power Grid environment tailored for RL evaluation.

B. The Power Grid Board Game

Power Grid[11] is a 3–6 player economic strategy game in which players compete to build a network of generators and supply electricity to the most cities. The environment we modeled uses the North American Deluxe Edition (2014), which divides the map into seven regions, only a contiguous subset of which is active depending on player count this is referred to as the Play Map as shown in Figure 1.

Each player begins with $50, a set of generator tokens, and an initial turn order determined randomly. The game contains a structured resource market consisting of four fuel types; coal, natural gas, oil, and uranium with each resource having a fixed global cap of 27, 24, 20, and 12 units, respectively. Power plants are represented by a deck of 42 cards, from which eight are revealed to form the power plant market. The four lowest-valued plants constitute the current auction market, while the next four form the future market. A special Step 3 card is placed at the bottom of the deck and triggers late-game rule changes when revealed.

Each game round consists of five sequential phases: *Determine Player Order*, *Auction Power Plants*, *Buy Resources*, *Build Generators*, and *Bureaucracy*. Druing the *Determine Player Order* Player order is recalculated based on who controls the most cities, with ties broken by the highest-numbered power plant owned. During the *Auction Power Plants* phase, players bid on power plants from the current market shown in Figure 2.

A player may own at most three power plant cards. If a fourth plant is purchased, the player must immediately discard one of their existing three plants from the game. Once a player starts an Auction on a plant, all players have the option to increase the bid or pass once all players have passed then the auction is complete and the plant awarded to the higgest bidder. A player may only purchase one plant per round and excluded from subsequent auctions in the round once they have acquired a power Plant card . *Buy Resources* occur in reverse turn order as a catch-up mechanic; fuel prices rise as the market depletes, and players may store at most twice the resources required to operate their plants. In the *Building Generators* phase (also taken in reverse player order), players expand their generator network by paying the required city cost together with the connection costs needed to link the new city to their existing network. The initial generator may be placed in any available city on the play map, but all subsequent generators must be connected to the player's existing network an example is shown in Figure 4. In the *Bureaucracy* phase, players spend resources to power their generators, earn income based on the number of powered cities, restock the resource

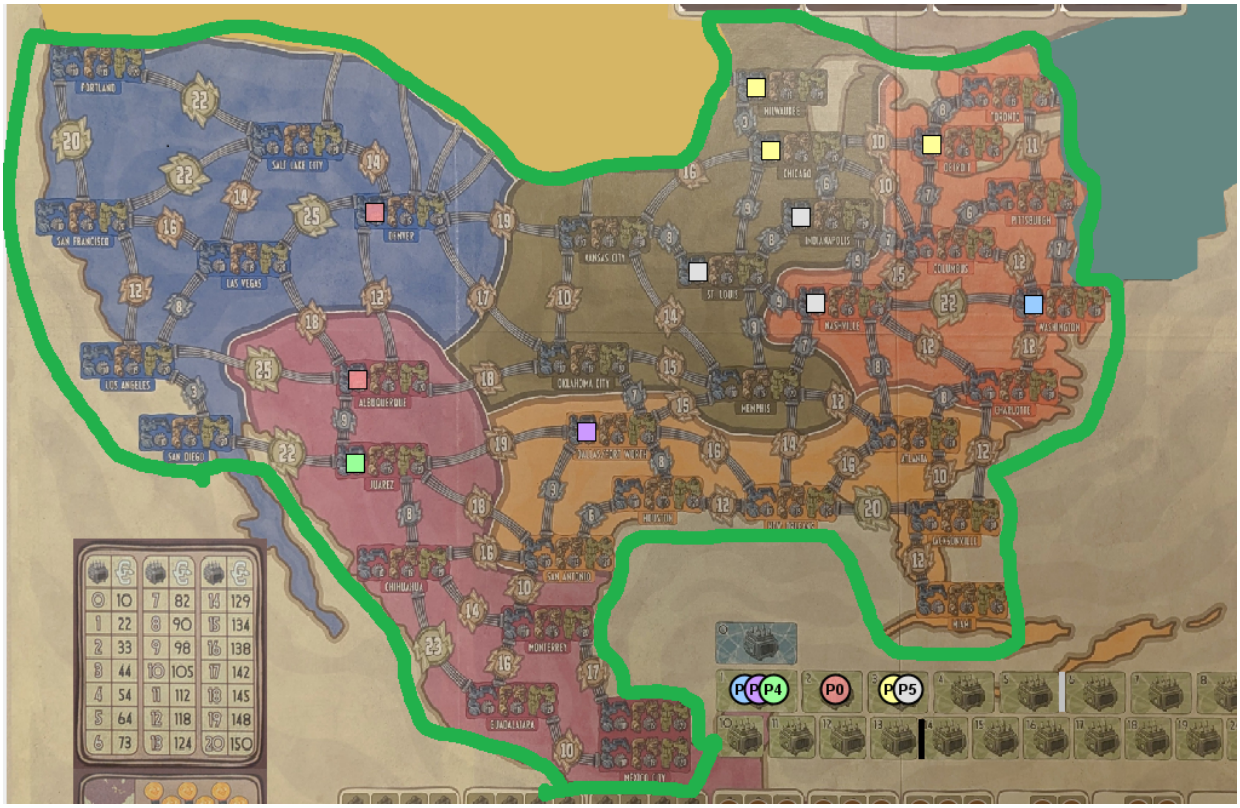


Fig. 1. Example Play Map for a six-player game. Five contiguous regions, outlined in green, constitute the Play Map, while the two inactive regions are Region 1 (Northeast) and Region 6 (West).

Fig. 2. The power plant markets. The current market, outlined in green, contains the plants available for auction, while the future market is outlined in red. When a plant is purchased, a new card is drawn from the deck and placed into the current market. The markets are then reordered in ascending order by plant number (top-left value on each card). When Step 3 begins, the two markets merge into a single current market, and the total number of market slots is reduced from eight to six. The Step Indicator is outlined in Blue there are 3 Steps total in the game.

market, and update the power plant market according to the current step rules shown in Figure 3.

The game progresses through three steps that modify placement rules and the size of the power plant market. Step 2 begins when any player builds a sixth generator, opening a second generator slot in each city at an increased cost. Step 3 begins when the Step 3 card is revealed, reducing the market to six plants and opening a third generator slot in each city. The game ends when any player builds enough generators to exceed the end-game threshold, which depends on player count. Victory is determined not by who triggers the end of the game, but by who can power the most cities in the final bureaucracy phase, with remaining cash breaking ties.

Power Grid is a fully observable game with minimal stochasticity. All player information; including money, owned power plants, resources, and network position is publicly

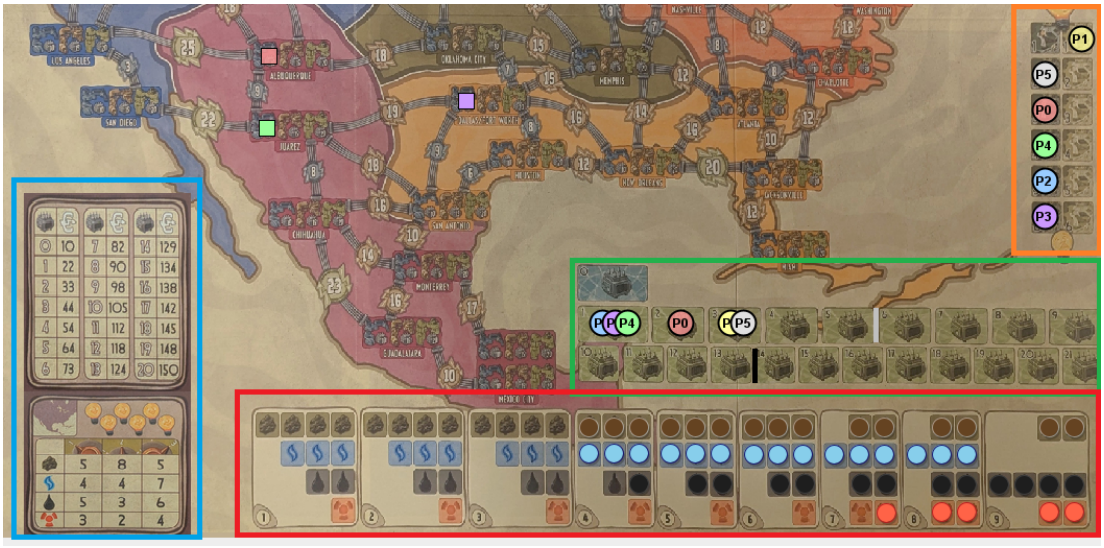


Fig. 3. Overview of the turn order tracker (orange), generator track (green), resource market (red), and resource refresh rates/income table (blue). The bottom row of the resource market corresponds to uranium; in this example, four units are available at a current price of 7. If one uranium is purchased, the price increases to 8 and three units remain. At the end of each round, resources are replenished according to the refresh rate table; under Step 1, uranium is replenished by three units. The turn order tracker indicates the current player order and which players have passed for the phase, with Player 1 shown as having passed. The generator track displays the number of generators owned by each player on the play map. The gray bar between spaces 5 and 6 marks the transition to Step 2, and the black bar indicates the game-end threshold.

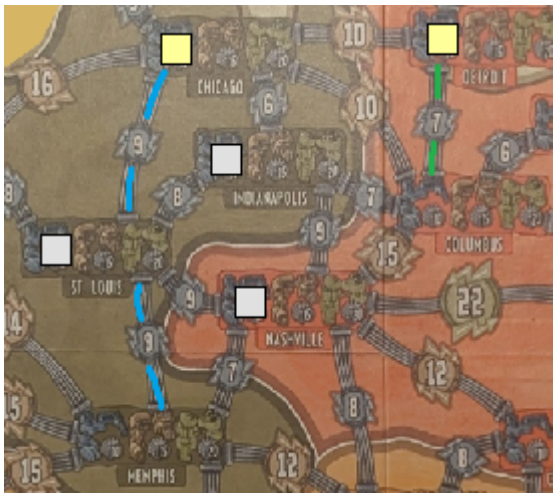


Fig. 4. Example city-building action. The Yellow player expands to Columbus (Green Path) at a total cost of $17, consisting of a $7 connection cost from the nearest city in their network (Detroit) and a $10 city build cost in Step 1. Yellow is blocked from building in St. Louis because the White player has already built there and Step 1 allows only one player per city. Building in Memphis would require the lowest cumulative connection cost to the existing network—$9 from Chicago to St. Louis and $9 from St. Louis to Memphis—plus the $10 build cost, for a total of $28 (Blue path).

visible at all times, and no hidden hands or private cards exist. The only source of randomness is the shuffled power plant deck. However, this randomness is heavily constrained by the structure of the power plant market. At all times, the four lowest-ranked plants form the *current market* and the next four form the *future market*. During the early game (prior to Step 3), the highest card in the future market is discarded to the bottom of the deck during the *Bureaucracy* phase, ensuring that late-game plants enter the market near ascending numerical order. When a plant is purchased, a new one is drawn and the markets are re-sorted to maintain the ordering rule. This mechanism significantly reduces randomness by controlling when stronger plants become available, making the progression of the market similar between games despite the initial deck shuffle.

III. Environment Methodology

Power Grid can be modeled as a Markov Decision Process (MDP). Unlike many other board games, the full game state is observable; all game state information is publicly visible. The only hidden element is the shuffled deck of power plant cards. In this section, we describe the core components of the environment used for reinforcement learning: the observation space, the action space, and the reward design. The entire environment is implemented using the TAG framework and wrapped with PyTAG for Python-based RL training.

A. Modeling Power Grid

1) MDP Formulation: The Markov Decision Process (MDP) of Power Grid (PG) can be modeled as a tuple $\mathcal{G} = (P, S, A, L, T, R, W, s_0)$. $P = \{p_1, p_2, \ldots, p_\rho\}$ is the set of players. S is the set of all possible game states, including player money and resources, owned cities, owned power plants, the resource market, the Current and Future plant markets, the remaining power plant deck, the current game phase, and turn order. A is the set of all possible actions, spanning all phases of the game (e.g., starting an auction, bidding, passing, buying resources, building cities, and running power plants). $L(p, a, s)$ is a legality function which returns whether action a is a legal action for player p in state s. In

Power Grid, L depends strongly on the current phase of the round, since only a restricted subset of actions is available in each phase. $T(s, a)$ is the transition function, mapping a state s and action a to the next state s'. Transitions are mostly deterministic, except for interactions with the Power Plant Market, which introduce stochasticity. $W(s)$ is the winner-determination function, applied in terminal states to identify the player (or players) with the highest performance according to the standard Power Grid rules (number of powered cities, with money used as a tiebreaker). Finally, s_0 is the initial state, corresponding to the setup after shuffling the plant deck, initializing the markets, and establishing the initial player order.

B. State Space

We estimate a conservative lower bound on the Power Grid state space by considering several major components for a 3-Player Game more higher player counts are more complex:

- **City ownership:** Each region has at least 6 cities with a three player play map being made up of 3 regions with 17 possible play maps as shown in tabble XI Each city has 8 possible ownership permutations, yielding $8^{(6\times3)} \times 17 \approx 3.06 \times 10^{17}$ possible configurations.
- **Resource market:** With quantities ranging from 0–27 coal, 0–24 gas, 0–20 oil, and 0–12 uranium, the market contributes $(27+1)(24+1)(20+1)(12+1) = 28\times25\times21\times13 \approx 2\times10^5$ states.
- **Power plant market:** The 8 visible plants selected from 42 total contribute $\binom{42}{8} \approx 1.2 \times 10^8$ configurations.
- **Power plant ownership:** Each player can hold up to three distinct plants from the 42-card deck, yielding approximately $\left(\sum_{k=0}^{3}\binom{42}{k}\right)^3 \approx 1.9\times10^{12}$ possible plant-ownership configurations.
- **Turn order:** Player ordering contributes $3! = 6$ possibilities.
- **Game phase:** The game progresses through 3 steps, each containing 4 phases (turn order determination, auction, building, and bureaucracy), contributing $3 \times 4 = 12$ discrete states.

Treating these as independent (a simplifying assumption, since plant ownership affects market state), the product yields approximately $10^{17}\cdot10^5\cdot10^8\cdot10^{12}\cdot10\cdot10 \approx 10^{44}$ states. This represents a lower bound, as it omits player resource storage, player money, and the additional permutations available with larger player counts. Nevertheless, this estimate places Power Grid at roughly the same state complexity as chess, which has approximately 10^{47} states [12].

C. Observation Space

At each decision point, an agent receives a structured observation vector that summarizes both player-specific information and global game state information. There is no player hidden information. The observation is decomposed into two major components:

TABLE I
PLAYER OBSERVATION VARIABLES AND NORMALIZATION

Symbol	Description	Normalization
m_i	Money of player i	500
c_i	Coal held by player i	9
g_i	Gas held by player i	9
o_i	Oil held by player i	9
ur_i	Uranium held by player i	6
cities_i	Number of cities owned by player i	$N_{\text{city}}^{\max}$
cap_i	Generation capacity of player i	20
inc_i	Income received by player i	150
$P_i^{(j)}$	j-th power plant owned by player i ($j = 1, 2, 3$)	50

- **Player Observation**, which encodes all features associated with each individual player (e.g., resources, income, capacity, and owned power plants).
- **Global Observation**, which captures all shared aspects of the environment, including the board topology, region configuration, power plant markets, resource market, turn and round order, and the number of cities powered in the previous turn.

This decomposition allows the agent to reason about both its own economic position and the strategic landscape of the full game state. Each subcomponent is normalized and encoded into a fixed-length vector so that the total observation dimension remains constant across different player counts and board configurations. The remainder of this section details the construction of each observation subvector and its contribution to the overall state representation.

1) Player Observation: Each agent receives a player-centric observation vector Z^{player} composed of normalized features describing all players in the game. For an agent with player index p, players are ordered *relatively*, so that the feature block h_0 corresponds to player p, h_1 to player $(p+1) \bmod \rho$, and so forth, where ρ is the number of players.

For each player i, we construct an 11-dimensional feature vector

$$h_i = \left[m_i,\ c_i,\ g_i,\ o_i,\ ur_i,\ \text{cities}_i,\ \text{cap}_i,\ \text{inc}_i,\ \text{P}_i^1,\ \text{P}_i^2,\ \text{P}_i^3\right],$$

where each component is normalized by a fixed upper bound established at initialization based on player count or a game constant according to

$$x^{\text{norm}} = \frac{x}{x_{\max}}.$$

The full player observation for one agent is the concatenation:

$$Z^{\text{player}} = [h_0, h_1, \ldots, h_{\rho-1}] \in \mathbb{R}^{11\rho}.$$

In the *player-agnostic* version, we fix $\rho_{\max} = 6$ and pad any unused player slots with zeros, yielding a constant-size vector

$$Z^{\text{player}} \in \mathbb{R}^{11\rho_{\max}}.$$

This ensures that the observation dimension remains stable across different player counts.

TABLE II
CITY OBSERVATION VARIABLES AND NORMALIZATION

Symbol	Description	Normalization
$\mathbf{s}_i$	One-hot slot vector for slot i	—
u	City ID	$N_{\text{city}}^{\max}$
w_{uv}	Cost between city u and neighbor v	$w_{\max} * 0.8$
$\mathbf{c}_u$	Static city vector	
$\mathbf{x}_u$	Full feature vector for city	

2) Board Structure: The board can be represented as an undirected graph in which nodes correspond to cities and edges represent the monetary cost of expanding from one city to another. Each city belongs to one of seven predefined regions. At the start of each game, a random subset of contiguous regions is selected to determine the playable map. The number of regions included depends on the total number of players, with larger player counts requiring a larger subset of regions.

The occupation of each buildable city slot is encoded as a one-hot vector $\mathbf{s}_i \in \{0,1\}^3$, corresponding to [mine, opponent, empty]. Since each city contains three such slots, its occupation state is represented by $[\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3]$. This occupation encoding is concatenated with a static, city-specific vector $\mathbf{c}_u$ that captures the structural properties of city u, including its unique identifier and the connection costs to all adjacent cities. Formally, if $v_1, \ldots, v_k$ denote the neighboring cities of u, then

$$\mathbf{c}_u = \big[\, u,\ w_{uv_1},\ w_{uv_2},\ \ldots,\ w_{uv_k} \,\big],$$

where u is the city index and w_{uv} is the normalized connection cost between cities u and v. Edges leading to cities in regions that are not in play are assigned a normalized cost of 1.0 to ensure they are treated as inaccessible; conversely, edges for cities in active regions are scaled by an additional factor of 0.8 after normalization by $w_{\max}$. The complete feature vector for city u is therefore

$$\mathbf{x}_u = \big[\, \mathbf{s}_1,\, \mathbf{s}_2,\, \mathbf{s}_3,\, \mathbf{c}_u \,\big].$$

Finally, a one-hot region-activation vector is included to indicate which of the seven predefined regions are part of the current game setup. This region vector is concatenated with every city feature vector, allowing the final board representation to encode both local city features and global region availability.

3) Power Plant Market: The Power Grid current plant and future plant market is represented by a normalized vector of size 8 that contains each card's unique id u normalized by Id max which is the constant 50 .

4) Resource Market: The resource market is represented as a four-dimensional vector containing the current quantities of coal, gas, oil, and uranium available for purchase. Each entry is normalized by the maximum supply of its corresponding resource: 27 for coal, 24 for gas, 20 for oil, and 12 for uranium.

5) Turn Order & Round Order: The *turn order* is represented as a fixed-length vector whose size equals the maximum number of players. Each index corresponds to a specific player ID, and the value stored at that index indicates that player's rank in the turn sequence. A value of 1 denotes the first player to act, while 0 denotes the last. For example, if Player 0 acts first, then index 0 of the vector stores a 1; if Player 2 is last, then index 2 stores a 0. Turn order remains static throughout an entire round.

The *phase order* vector has the same structure but is updated dynamically at the beginning of each phase. When a player passes, the game engine assigns them a value of -1, which we normalize to 0 in the observation space to indicate that the player has passed and is therefore placed at the end of the order for that phase.

6) Cities Powered: The *cities-powered* vector represents the number of cities powered by each player in the previous turn. Each value is normalized by the maximum number of cities powered by any player (or by 1 if this maximum is zero), yielding a score in the range $[0, 1]$. Higher values indicate greater efficiency in converting resources into powered cities during that turn.

TABLE III
POWER GRID ACTION SPACE BY PHASE. ACTION COUNTS ARE SHOWN FOR EACH PHASE AND FOR THE TOTAL ACTION SPACE. HERE, s IS A POWER-PLANT SLOT (0–2), i A RESOURCE-TYPE INDEX, k A POWER-PLANT ID, AND u A CITY ID.

Phase	Action Type	Action Count
Determine Player Order	None	0
Auction Power Plants	auction_plant_k	42
	increase_bid	1
	pass_bid	1
	discard_s	3
Buy Resources	buy_Resource_i	33
Build Generators	build_city_u	47
Bureaucracy	run_k_Resource_i	50
All Phases	pass_round	1
Total		**178**

The action space in *Power Grid* differs notably from those found in most board games commonly used in reinforcement-learning research. Unlike environments such as *Ticket to Ride*, *Go*, or *Chess*, *Power Grid* employs a phase-based round structure in which the set of legal actions changes dynamically as the game progresses. Within a single round, each phase presents a distinct subset of legal actions from the previous phase. This section outlines the phases in which decision-making occurs and describes the corresponding action types available within each phase.

7) Auction Power Plants: The Auction Power Plants phase contains the action to place a power plant from the current market up for bidding. When a player initiates an auction, the game engine enters a bidding subphase that iterates over all players who remain eligible to bid— those who have not passed in the current auction and who have not already won a plant this round. Bidding continues until all but one player have passed, at which point the remaining player wins the auction and acquires the plant. If the acquired plant would

become the player's fourth, the game presents three discard actions corresponding to the three power-plant slots in the player's inventory.

Because this bidding procedure operates within a subphase, multiple distinct action types can appear in this phase, not all of which are legal at every decision step. This is why Table III lists several action types specific to the Auction Power Plants phase.

8) Buy Resources: The Buy Resources phase allows players to purchase fuel for their power plants. The game engine determines the legal purchases available to a player based on their remaining money and the storage capacity of their owned plants. It then enumerates all valid combinations of resource type and quantity that the player can both afford and store. After a purchase, the engine recomputes the valid purchase set, and this process continues until the player either exhausts all legal options or elects to pass the phase.

As shown in Table III, this phase contains 33 possible actions. This arises from the structure of the resource market: coal, gas, and oil each have up to nine purchase amounts; two hybrid plants introduce three additional combinations; another hybrid plant contributes four more; and uranium provides six purchase amounts.

9) Build Generators: The Build Generator phase allows players to expand their network by placing new generators on the map. The game engine determines all legal build actions by performing a *breadth-first search* (BFS) originating from the player's existing network. For each reachable city, the engine accumulates both the connection cost required to extend the network and the construction cost associated with any available city slots. Cities whose total cost exceeds the player's money or that contain no valid slots—due to occupation rules—are excluded from the action set.

Because the board contains 47 cities, the theoretical maximum number of build actions is also 47. However, the legal set in any game state is always smaller, as only the subset of cities that make up the play map are available. During the first turn, before any generators have been built, BFS does not apply and the player may build in any city within the active regions, paying only the city construction cost.

10) Run Power Plants: The Run Power Plants phase allows players to choose which of their owned power plants to activate. For most plants, each action corresponds directly to a plant's card ID. However, the presence of three hybrid plants—each of which may be fueled by any combination of oil and gas—expands the action space beyond the 42 plant IDs. To account for the fuel-combination choices associated with these hybrids, the environment increases the action space to 50 distinct run actions.

To determine which run actions are legal, the game engine checks each plant's resource requirements and verifies that the player has sufficient fuel. Only plants for which the player meets the full input requirements, including hybrid combinations, are included in the valid action set.

11) Action Masking and Phase-Dependent Valid Actions: We structure the discrete action space using a *Conditional Action Tree* (CAT), following the formulation in [13]. The CAT encodes the full hierarchical structure of all possible actions in the game and is fixed when the environment is initialized. At each decision step, the environment uses the CAT as a static template and prunes branches corresponding to actions that are not returned as legal by the forward model [4]. The remaining leaves of the pruned tree form the set of actions that are legal in the current state.

To interface this hierarchical representation with a flat discrete action space, we linearize the leaves into a binary action mask

$$m \in \{0,1\}^{|\mathcal{A}|},$$

defined over the fixed global action set $\mathcal{A}$. The policy network outputs logits

$$\ell \in \mathbb{R}^{|\mathcal{A}|}.$$

Before applying the softmax, invalid-action masking is enforced by replacing the logits of all illegal actions with a large negative constant:

$$\ell_a \leftarrow -10^8 \quad \text{for all } a \text{ such that } m_a = 0.$$

This guarantees that the resulting action distribution assigns zero probability to illegal actions and ensures that the agent selects only valid moves in the current state [5].

Using a static CAT for pruning provides efficient mask generation: high-level phase and subphase constraints remove entire branches of the tree, eliminating large sets of incompatible actions without requiring enumeration of the full action space at every timestep.

D. Reward Structure

Prior work using TAG environments typically relied on a binary win–loss reward structure [5]. While effective for short games, such terminal-only rewards provide weak learning signals in long, multi-phase environments such as Power Grid. To address this, we introduce an intermediate reward computed during the Bureaucracy phase.[1]

$$r_t = \begin{cases} \alpha \dfrac{\text{generators}_t}{\text{GeneratorMax}} + \beta \dfrac{\text{income}_t}{\text{IncomeMax}}, & \text{if phase} = 5, \\ 0, & \text{otherwise.} \end{cases}$$

Intermediate rewards are provided only during Phase 5 (Bureaucracy), while Phases 1–4 yield zero reward. The reward consists of two normalized components: the number of generators powered and the income earned at time t. The constants GeneratorMax and IncomeMax scale these values to the range $[0, 1]$, ensuring that the weights α and β cleanly determine their relative contributions, with $\alpha+\beta = 1$. Because income may decrease between rounds, this reward formulation provides a meaningful, non-monotonic signal that encourages stable and efficient network expansion throughout the game.

[1]For *Power Grid Deluxe*, GeneratorMax $= 21$—the maximum number of generators a player may power—and IncomeMax $= 150$, the highest possible income value in the Bureaucracy phase.

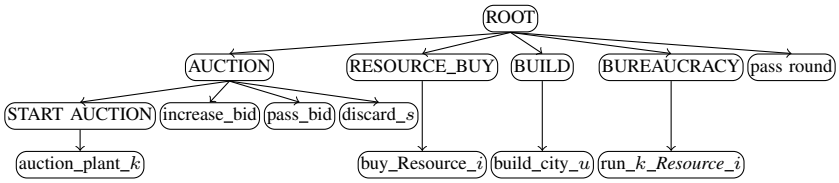


Fig. 5. Power Grid action tree structure. Actions are leaf nodes (lowercase spelling), internal nodes Present Phase and Sub Phases (uppercase spelling).

IV. Experimental Methodology

A. Algorithms/Opponenets

1) Random: The Random agent is implemented directly by the TAG engine. On each decision step it uniformly samples an action from the set of legal actions. This approach requires virtually no computation and serves as a baseline for comparison against more informed decision-making algorithms.

2) OSLA: The One-Step Look Ahead (OSLA) agent evaluates every legal action by applying it once to the forward model and selecting the successor state with the highest immediate reward. Although simple, this greedy strategy performed worse than the Random agent across all evaluated player counts. This outcome suggests that Power Grid's multi-phase structure rewards longer-horizon planning, and that optimizing only for immediate gains often leads to poor long-term outcomes. For this reason it was not used in training any RL models.

3) Monte Carlo Tree Search (MCTS): For baseline statistical foward planning opponent, we use the default MCTS implementation provided by the TAG framework. MCTS incrementally builds a game tree from the current state using repeated selection–expansion–evaluation–backup cycles[14]. TAG employs a closed-loop variant of UCT in which nodes store full game states and newly expanded nodes are evaluated immediately using a state heuristic, without performing random rollouts[4].

For Power Grid, we utilize a custom leader-based state heuristic that guides the search. The heuristic evaluates a player's position using five component: number of connected cities, available money, income level, total power-plant capacity, and fuel resources relative to the requirements of the player's plants. For each component the player's value is compared to the current leader, and an asymmetric saturating function rewards being ahead and penalizes falling behind. The shaped components are combined into a weighted sum and normalized to produce a score in $[0, 1]$, which is used as the node value during MCTS search.

4) Proximal Policy Optimization: Proximal Policy Optimization (PPO) is an on-policy reinforcement learning algorithm that constrains policy updates using a clipped surrogate objective, preventing destructively large parameter shifts while maintaining good sample efficiency [15]. PPO has demonstrated strong performance across a wide range of continuous and discrete control tasks, making it a natural baseline for complex board-game environments.

In this work, we employ the Maskable PPO implementation provided by Stable-Baselines3 [16], which extends standard PPO to handle dynamically changing action spaces via invalid-action masking. In standard PPO, the policy network outputs a probability distribution over all actions, implicitly assuming that every action is legal in every state. Maskable PPO instead accepts a binary action mask

$$m \in \{0, 1\}^{|\mathcal{A}|},$$

where $m_a = 0$ indicates that action a is illegal in the current state. Before applying the softmax over action logits, all logits corresponding to illegal actions are replaced with large negative values, ensuring that the policy assigns zero probability to invalid moves.

This masking mechanism is essential for *Power Grid*, where action legality depends on the current phase, player resources, board position, and auction dynamics. The set of legal actions changes at nearly every decision step—for example, during the Auction phase only unowned plants can be bid on, while during the Build phase only cities reachable from the player's existing network are valid construction targets. The TAG forward model automatically generates these masks by pruning the conditional action tree, and PyTAG passes them directly to the Maskable PPO policy at each timestep.

B. Training Setup

All experiments were conducted using a total of five million training time steps. We utilized a parallelized PPO training setup with eight vectorized environments (`n-envs = 8`). Each environment executed 512 steps per rollout, producing a combined batch of $512 \times 8 = 4096$ transitions per update. The core training hyperparameters were:

- **n-steps:** 512
- **batch size:** 1024
- **n-epochs:** 4
- **discount factor** (γ): 0.995
- **GAE** λ: 0.97
- **entropy coefficient:** 0.005
- **value function coefficient:** 0.7
- **clip range:** 0.15
- **learning rate:** 3×10^{-4}
- **maximum episode length:** 300 steps
- **total timesteps:** 5,000,000
- **policy architecture:** two-layer MLP with 256 hidden units per layer (256×256)

Due to the long training horizon, only limited hyperparameter tuning was feasible. We therefore performed a coarse

hyperparameter sweep over shorter training runs (1,000,000 environment steps) to identify reasonable settings before full-scale training.

Specifically, we evaluated the following values:

- policy/value MLP hidden layers: $\pi = [64, 64], V = [64, 64]$ vs. $\pi = [256, 256], V = [256, 256]$
- discount factor $\gamma \in \{0.99, 0.995\}$
- GAE parameter $\lambda \in \{0.90, 0.95, 0.97\}$

All other PPO hyperparameters were held fixed.

The evaluation procedure consisted of several stages. First, a PPO agent was trained against fully random opponents across all player counts. Next, a single MCTS opponent was introduced while the remaining opponents remained random; this was repeated for each player count. In a subsequent set of experiments, agents trained against MCTS were evaluated against agents trained exclusively against random opponents in order to assess generalization and robustness.

All evaluations were performed deterministically over 1000 episodes to ensure stable and reproducible results.

V. Results

All results were collected using 1000 deterministic evaluation episodes. For all comparisons based on phase-action statistics, models were evaluated against opponents trained on the same player count.

A. Statistical Forward Planning Baselines

TABLE IV
STATISTICAL FORWARD PLANNING MODEL COMPARISON

Players	MCTS	OSLA	Random
3	82.3%	0.7%	17.0%
4	70.5%	0.5%	14.4%
5	64.4%	0.2%	11.8%
6	54.4%	0.4%	11.3%

Results were obtained in the TAG framework using 1000 self-play episodes per player count, with no reinforcement learning agents involved.

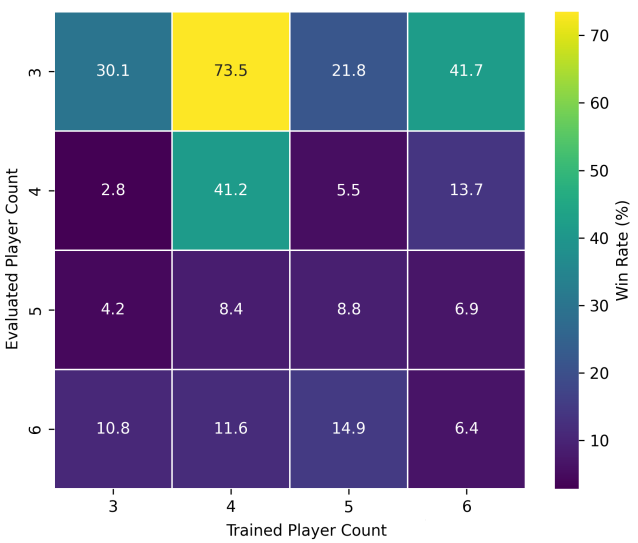


Fig. 6. Win-rate heatmap for MCTS evaluated against PPO agents trained using MCTS opponents.

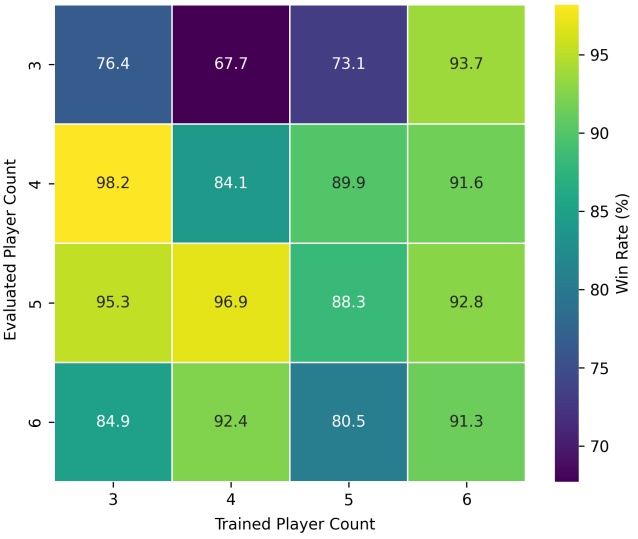


Fig. 7. Win-rate heatmap for PPO agents trained against random opponents across player counts.

TABLE V
POWER PLANT FUEL-TYPE USAGE UNDER RANDOM OPPONENTS.

Type	Deck	Obs	Exp	Obs%	Exp%	StdResid
Coal	12	24465	53938.6	12.96	28.57	−126.9
Gas	8	58599	35959.0	31.04	19.05	119.4
Oil	7	14231	31464.2	7.54	16.67	−97.2
Hybrid	3	3901	13484.6	2.07	7.14	−82.5
Uranium	5	12309	22474.4	6.52	11.90	−67.8
Renewable	7	75280	31464.2	39.88	16.67	247.0

Totals reflect data aggregated from agents trained and evaluated on the same player count. $\chi^2 = 112{,}223.5$, $p < 10^{-6}$, $n = 188{,}785$.

B. PPO Against Random Opponents

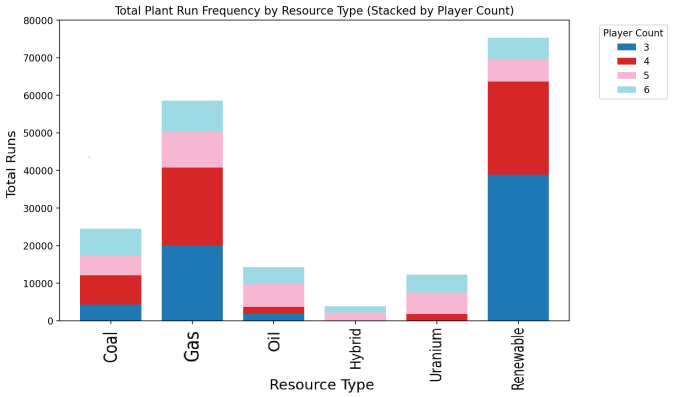


Fig. 8. Plant run frequency by resource type for PPO trained against Random opponents.

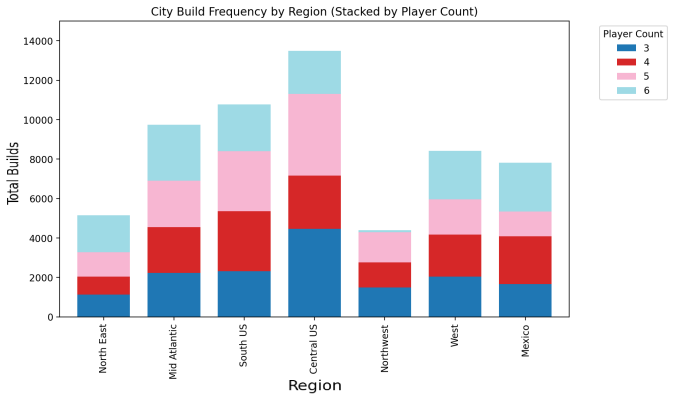


Fig. 9. City build frequency by region under Random opponents.

C. PPO Against MCTS Opponents

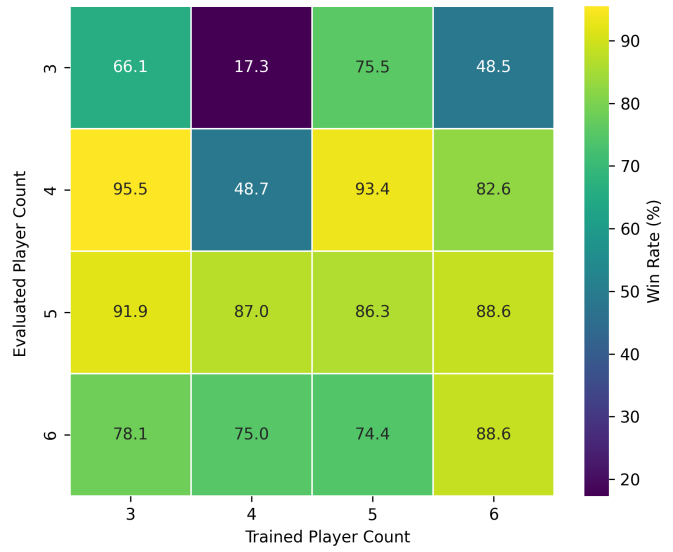


Fig. 10. Win-rate heatmap for PPO agents trained against MCTS opponents across player counts.

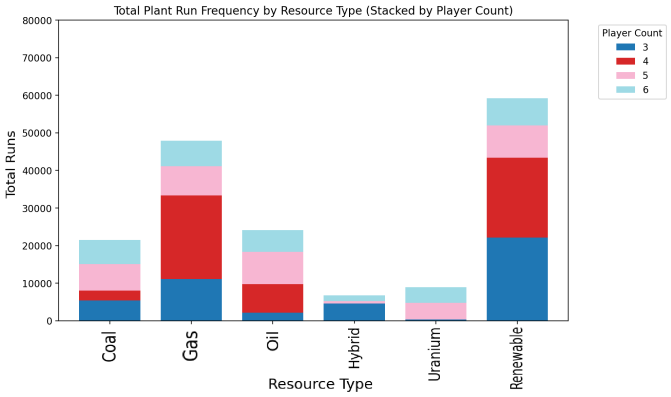


Fig. 11. Plant run frequency by resource type for PPO trained against MCTS opponents.

TABLE VI
POWER PLANT FUEL-TYPE USAGE UNDER MCTS OPPONENTS.

Type	Deck	Obs	Exp	Obs%	Exp%	StdResid
Coal	12	21505	48095.7	12.78	28.57	−121.3
Gas	8	47852	32063.8	28.43	19.05	88.2
Oil	7	24104	28055.8	14.32	16.67	−23.6
Hybrid	3	6790	12023.9	4.03	7.14	−47.7
Uranium	5	8892	20039.9	5.28	11.90	−78.8
Renewable	7	59192	28055.8	35.16	16.67	185.9

$\chi^2 = 66{,}066.4$, $p < 10^{-6}$, $n = 168{,}335$.

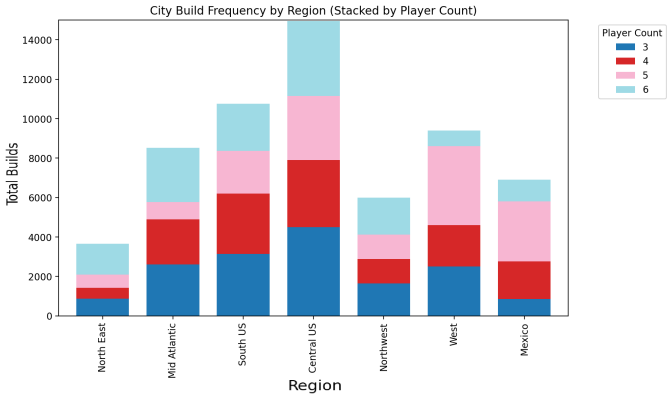


Fig. 12. City build frequency by region for PPO trained against MCTS opponents.

D. Cross-Opponent Generalization

TABLE VII
WIN RATES FOR PPO AGENTS TRAINED AGAINST MCTS AND RANDOM OPPONENTS.

Players	MCTS-trained PPO	Random-trained PPO	Random baseline
3	51.6%	35.3%	13.1%
4	47.8%	46.9%	2.65%
5	56.2%	40.9%	6.66%
6	29.2%	62.9%	1.97%

VI. Discussion

The Power Grid environment presents a challenging strategic landscape for PPO, producing distinct behavioral patterns across different player counts and opponent types. As shown in the win-rate heatmaps in Figures 7 and 10, a consistent trend emerges: PPO tends to perform better as the number of players increases, while the MCTS baseline dominates at lower player counts. The 4-player configuration is a notable exception, producing mixed and sometimes counterintuitive outcomes, suggesting that this particular setting may introduce unique competitive dynamics.

A similar divergence appears in episode length. For the Random baseline, Table IX shows that average episode length decreases as the player count increases. This trend does not hold against the MCTS opponent, where the step counts remain comparatively stable across player counts (Table X). These differences suggest that PPO adapts its play style to the predictability and strength of its opponents. Against all-random opponents, PPO can safely prolong the game to accumulate additional intermediate rewards with minimal risk of falling behind. In contrast, when facing an opponent guided by a leader-based heuristic, PPO appears to favor concluding the game more quickly, likely to prevent the MCTS agent from leveraging its stronger tactical evaluations. This variation in episode length provides evidence of strategic adaptation by PPO relative to opponent type.

The unusually high variability observed in the 3-player evaluations can be explained by underlying map diversity. As shown in Table XI, the 3-player configuration has fifteen possible region combinations, compared to twelve for the 5–6 player case. Importantly, all smaller-player maps are strict subsets of the larger-player maps, meaning that agents trained in larger games are exposed to a more complete range of spatial configurations. This broader exposure may explain why larger-player PPO models generalize more effectively across the full map space, whereas agents trained in smaller games encounter more fragmented and idiosyncratic region layouts.

City-building patterns support this interpretation. Both PPO and MCTS consistently favor centrally located regions, as shown in Figures 9 and 12. Region 4 (Central United States) appears frequently in valid configurations, but even after accounting for this, it is still disproportionately preferred. Table XII shows that Region 4 offers many average city connections, reducing the risk of being blocked by opponents while providing strong expansion flexibility. Region 3 is the second most commonly selected despite only average connectivity, suggesting that both agents have learned that it still provides competitive long-term positioning.

Resource selection behavior follows a similar pattern. Across all experiments, both agents show a strong preference for Natural Gas and Renewable plants (Figures 8 and 11). Natural Gas offers the highest late-game refresh rate (Table XIII), making it a reliable long-term fuel source, while Renewables avoid resource costs entirely. Although renewable plants typically have lower capacity, the absence of resource constraints appears to outweigh this limitation. Coal, by contrast, becomes one of the scarcest resources by Step 3 despite its high initial availability, making it far less attractive for sustained planning. The chi-square analyses in Tables V and VI confirm that plant-type usage differs significantly from expected distributions, indicating that PPO adopts consistent preferences rather than sampling plants uniformly.

To further analyze plant acquisition behavior, we compare how often PPO initiates an auction on a given plant versus how often it ultimately runs that plant, as shown in Figures 17 and 18. Plants with **high auction frequency but low run frequency** indicate that the agent frequently initiates auctions it does not eventually win. This suggests that auction initiation may serve as a relatively low-cost exploratory action or as a mechanism for advancing market turnover, rather than as a strong indicator of preference for that specific plant. Conversely, plants with **low auction frequency but high run frequency** tend to be acquired after other players initiate the auction, implying that the agent's bidding behavior is more reactive than proactive for these particular plant types. Although each player count exhibits distinct patterns, the qualitative relationship between auction and run frequency is consistent across Random and MCTS opponents. The three-player cases (Figures 17a and 18a) show the greatest deviation from the equal-magnitude line, suggesting that strategic variability is highest when the competitive landscape is the least diffuse.

The superior performance of MCTS at low player counts appears to be related to the structure of its leader-based heuristic. With fewer opponents, leadership metrics such as money, cities, and generation capacity tend to remain more stable, allowing MCTS to produce more consistent evaluations of the game state. As player count increases, these metrics become more volatile, with different players leading different components of the heuristic at different times. This added variability may reduce the reliability of the heuristic and make long-term planning more difficult for MCTS. PPO, by contrast, benefits from function approximation, which may help it adapt to more complex and dynamic competitive settings. This can observed in comparing MCTS with the Random and OSLA baselines in Table IV. MCTS outperforms both across all player counts but shows its largest margins in the low-player settings, suggesting that it functions as a more informative training opponent when the environment is less congested. This observation is consistent with the results in Table **??**, which compare the best PPO models trained against MCTS and Random opponents for a given player count. In these experiments, the MCTS-trained PPO models perform better in all player counts except the six-player case, where the Random-trained model achieves superior results. Taken together, these findings suggest that leader-based heuristics may be more suitable as training opponents in low-player-count environments, whereas randomized opponents may provide a more effective training opponent as player count increases

VII. Conclusion

This work introduced Power Grid as a multi-player reinforcement learning environment, implemented in the Tabletop Games (TAG) framework and accessible via PyTAG. We formalized the game as an MDP with a structured observation space supporting variable player counts, a phase-dependent action space managed through conditional action trees, and reward shaping tailored to the game's multi-phase economic structure. Baseline evaluation of PPO agents against Random and MCTS opponents across 3–6 player games revealed three key findings. First, MCTS outperforms PPO in low-player settings where leader-based heuristics remain stable, while PPO scales better as player count increases and competitive dynamics become more volatile. Second, both learning and planning agents converge on similar strategic preferences—favoring renewable and gas plants, selecting centrally located regions, and exhibiting consistent auction behavior—despite different decision-making mechanisms. Third, PPO adapts episode length based on opponent strength, prolonging games against Random agents to accumulate intermediate rewards while ending games more quickly against MCTS. The primary contribution is the environment itself: a complex, reproducible benchmark that isolates challenges arising from large action spaces, long horizons, and multi-agent competition in a fully observable setting. By releasing the TAG implementation alongside baseline results, we provide infrastructure for systematic study of policy learning in phase-based strategy games. Future work can leverage this environment to investigate hierarchical policies that exploit phase structure, opponent modeling and adaptive play, curriculum learning across player counts, and transfer to other economic strategy games with similar mechanics.

References

[1] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, 01 2016.

[2] K. Sfikas and A. Liapis, "Collaborative agent gameplay in the pandemic board game," in *International Conference on the Foundations of Digital Games*, ser. FDG '20. ACM, Sep. 2020, p. 1–11. [Online]. Available: http://dx.doi.org/10.1145/3402942.3402943

[3] S. Yang, M. Barlow, T. Townsend, X. Liu, D. Samarasinghe, E. Lakshika, G. Moy, T. Lynar, and B. Turnbull, "Reinforcement learning agents playing ticket to ride–a complex imperfect information board game with delayed rewards," *IEEE Access*, vol. 11, pp. 60 737–60 757, 2023.

[4] R. D. Gaina, M. Balla, A. Dockhorn, R. Montoliu, and D. Perez-Liebana, "TAG: A Tabletop Games Framework," in *Experimental AI in Games (EXAG), AIIDE 2020 Workshop*, 2020.

[5] M. Balla, G. E. M. Long, D. Jeurissen, J. Goodman, R. D. Gaina, and D. Perez-Liebana, "Pytag: Challenges and opportunities for reinforcement learning in tabletop games," 2023. [Online]. Available: https://arxiv.org/abs/2307.09905

[6] P. Walker-Harding, "Sushi go!" 2013, board game.

[7] E. Lee, S. Small, and M. Inman, "Exploding kittens," 2015, board game.

[8] B. Faidutti and A. R. Moon, "Diamant," 2005, board game.

[9] S. Kanai, "Love letter," 2012, board game.

[10] J. J. Mogendorff, "Stratego," 1946, board game.

[11] F. Friese, "Power grid," 2004, board game.

[12] A. Atashpendar, T. Schilling, and T. Voigtmann, "Sequencing chess," *EPL (Europhysics Letters)*, vol. 116, no. 1, p. 10009, Oct. 2016. [Online]. Available: http://dx.doi.org/10.1209/0295-5075/116/10009

[13] C. Bamford and A. Ovalle, "Generalising discrete action spaces with conditional action trees," 2021. [Online]. Available: https://arxiv.org/abs/2104.07294

[14] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A Survey of Monte Carlo Tree Search Methods," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, p. TCIAIG.2012, Jan. 2012.

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

[16] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

Appendix

TABLE VIII
KEY TERMS USED IN *Power Grid*

Term	Definition
Round	A complete cycle of gameplay consisting of all five phases. There is no limit to the Number of Rounds but Victory is checked at the end of each Round.
Phase	A structured subdivision of a round. Each phase restricts which actions are legally available (e.g., Auction, Resource Buying, Building, Bureaucracy).
Turn	A single player's opportunity to act within a phase (e.g., bid, buy resources, build). Players may act multiple times per round across different phases.
Play Map	A subset of contiguous regions that form a valid board for a given player count .

TABLE IX
AVERAGE NUMBER OF STEPS PER EPISODE (MEAN $\pm$ STD) FOR PPO TRAINED AGAINST RANDOM OPPONENTS.

Evaluated \ Trained	**3P**	**4P**	**5P**	**6P**
3P	222.4 $\pm$ 31.4	220.0 $\pm$ 30.0	234.2 $\pm$ 31.1	184.8 $\pm$ 36.8
4P	128.2 $\pm$ 15.9	215.3 $\pm$ 28.0	179.8 $\pm$ 27.8	167.5 $\pm$ 25.5
5P	123.9 $\pm$ 16.9	136.7 $\pm$ 21.3	167.0 $\pm$ 26.6	151.5 $\pm$ 24.7
6P	120.5 $\pm$ 16.9	137.5 $\pm$ 22.3	151.8 $\pm$ 24.1	153.2 $\pm$ 22.1

TABLE X
AVERAGE NUMBER OF STEPS PER EPISODE (MEAN $\pm$ STD) FOR PPO TRAINED AGAINST MCTS.

Trained \ Evaluated	3P	4P	5P	6P
3P	174.4 $\pm$ 50.6	211.7 $\pm$ 32.2	190.2 $\pm$ 43.4	234.2 $\pm$ 31.4
4P	130.9 $\pm$ 14.6	204.0 $\pm$ 31.2	168.9 $\pm$ 25.9	199.1 $\pm$ 33.6
5P	123.5 $\pm$ 16.9	131.7 $\pm$ 20.8	173.5 $\pm$ 23.1	170.1 $\pm$ 26.2
6P	116.7 $\pm$ 16.2	131.2 $\pm$ 27.5	160.9 $\pm$ 23.8	157.3 $\pm$ 26.6

TABLE XI
VALID REGION COMBINATIONS FOR EACH PLAYER COUNT IN POWER GRID. VALIDITY IS DETERMINED BY SELECTING A SUBSET OF MAP REGIONS THAT FORMS A SINGLE CONTIGUOUS PLAY MAP ON THE UNDERLYING REGION ADJACENCY GRAPH.

3 Player	**4 Player**	**5 & 6 Player**
(1, 2, 3)	(1, 2, 3, 4)	(1, 2, 3, 4, 5)
(1, 2, 4)	(1, 2, 3, 7)	(1, 2, 3, 4, 6)
(2, 3, 4)	(1, 2, 4, 5)	(1, 2, 3, 4, 7)
(2, 3, 7)	(1, 2, 4, 6)	(1, 2, 3, 6, 7)
(2, 4, 5)	(1, 2, 4, 7)	(1, 2, 4, 5, 6)
(2, 4, 6)	(2, 3, 4, 5)	(1, 2, 4, 5, 7)
(2, 4, 7)	(2, 3, 4, 6)	(1, 2, 4, 6, 7)
(3, 4, 5)	(2, 3, 4, 7)	(2, 3, 4, 5, 6)
(3, 4, 6)	(2, 3, 6, 7)	(2, 3, 4, 5, 7)
(3, 4, 7)	(2, 4, 5, 6)	(2, 3, 4, 6, 7)
(3, 6, 7)	(2, 4, 5, 7)	(2, 3, 5, 6, 7)
(4, 5, 6)	(2, 4, 6, 7)	(2, 4, 5, 6, 7)
(4, 5, 7)	(3, 4, 5, 6)	(3, 4, 5, 6, 7)
(4, 6, 7)	(3, 4, 5, 7)	
(5, 6, 7)	(3, 4, 6, 7)	
	(3, 5, 6, 7)	
	(4, 5, 6, 7)	
Total: 15	**Total: 17**	**Total: 13**

TABLE XII
SUMMARY OF THE SEVEN REGIONS ON THE POWER GRID (NORTH AMERICA) MAP, INCLUDING TOPOLOGICAL STATISTICS AND THE PROBABILITY (IN PERCENT) THAT EACH REGION IS INCLUDED IN THE ACTIVE GAME CONFIGURATION FOR DIFFERENT PLAYER COUNTS

Region	**Name**	**# Cities**	**Total Cost**	**# Connections**	**Avg. Cost**	**3 Player**	**4 Player**	**5 & 6 Player**
1	North East	6	101	12	8.42	13.3%	29.4%	53.8%
2	Mid Atlantic	7	239	24	9.96	46.7%	70.6%	92.3%
3	South US	7	248	20	12.40	46.7%	58.8%	69.0%
4	Central US	7	276	25	11.04	73.3%	82.4%	85.0%
5	Northwest	7	208	14	14.86	33.3%	47.1%	62.0%
6	West	7	333	19	17.52	40.0%	52.9%	69.0%
7	Mexico	6	255	16	15.94	46.7%	58.8%	69.0%

TABLE XIII
RESOURCE REFRESH RATES FOR EACH PLAYER COUNT IN *Power Grid*, BROKEN DOWN BY STEP 1–3.

Resource	3 Players			4 Players			5 Players			6 Players		
	S1	S2	S3	S1	S2	S3	S1	S2	S3	S1	S2	S3
Coal	2	5	2	3	6	3	4	7	4	5	8	5
Gas	2	2	4	3	3	5	3	3	6	4	4	7
Oil	3	1	3	3	2	4	4	2	5	5	3	6
Uranium	2	1	2	2	1	2	3	2	3	3	2	4

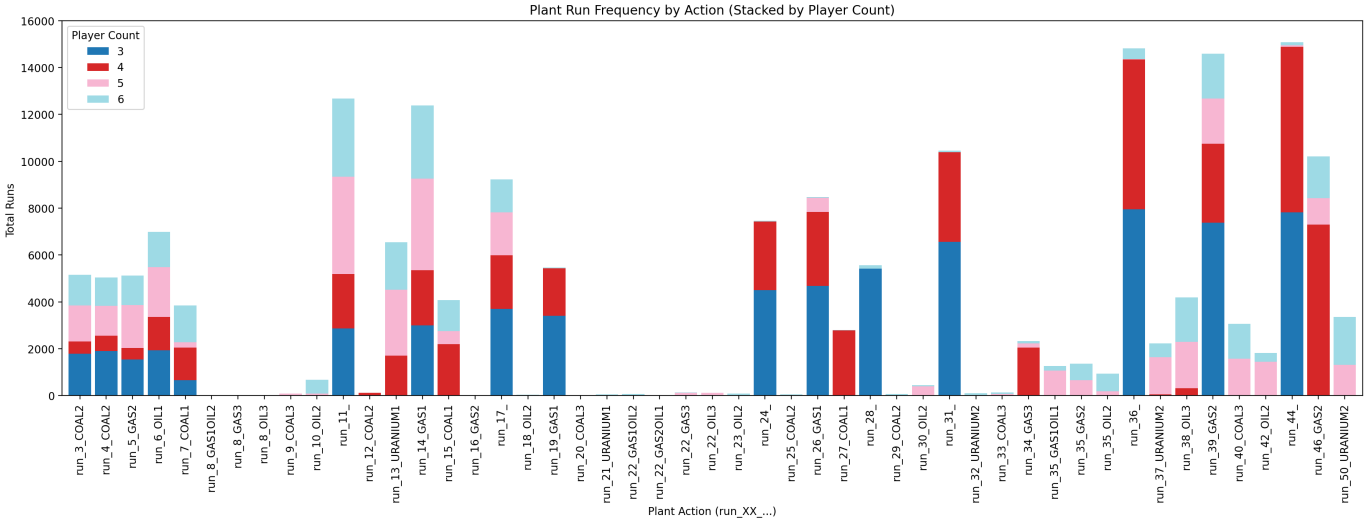


Fig. 13. Plant run frequency by individual plant action for the Random agent.

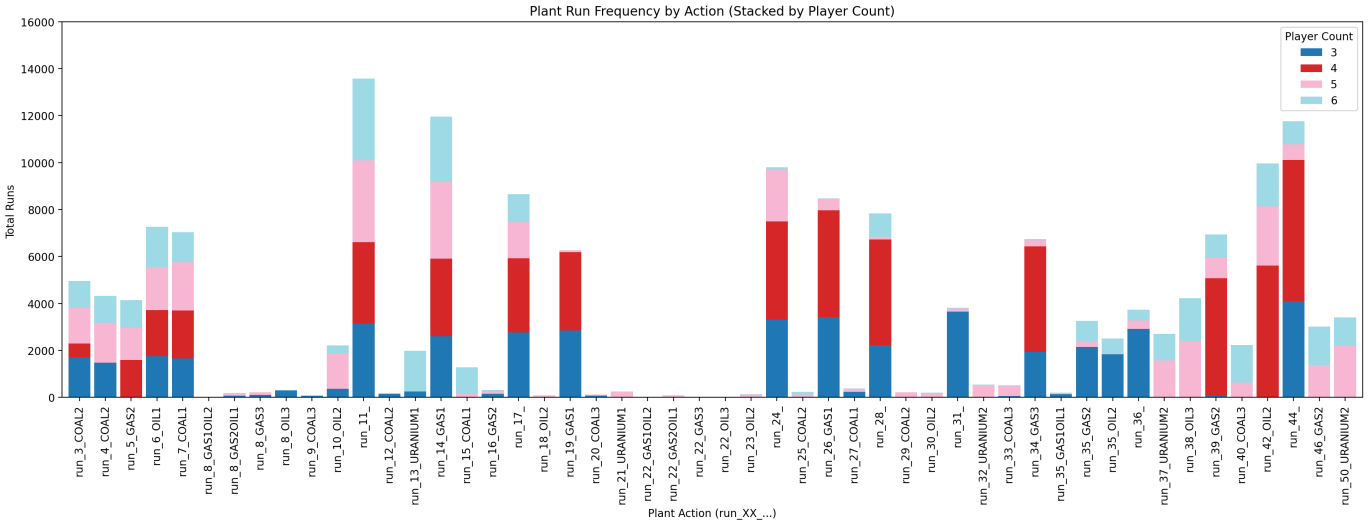


Fig. 14. Plant run frequency by individual plant action for the MCTS agent.

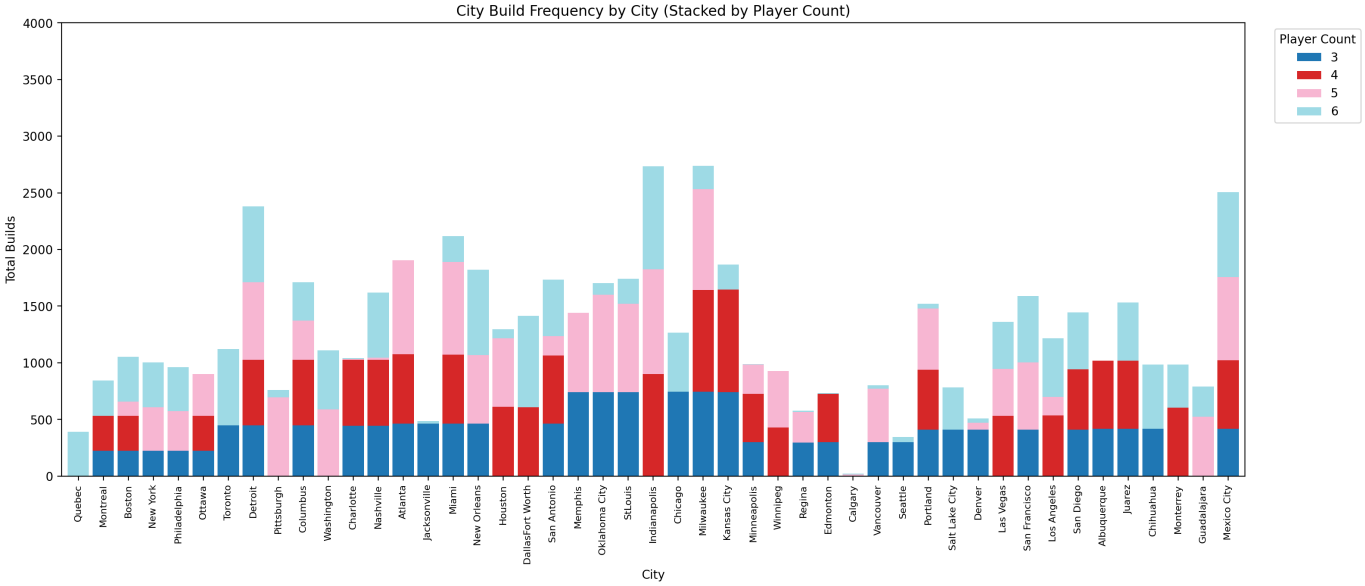


Fig. 15. City build frequency by city for the Random agent, aggregated across all players.

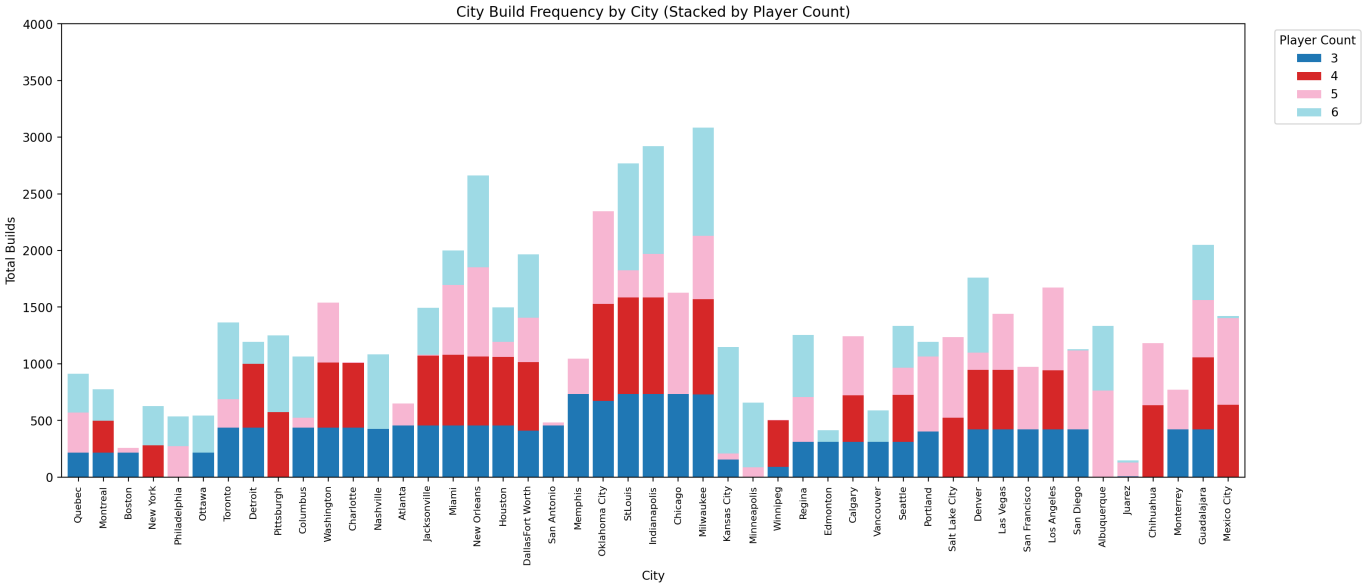
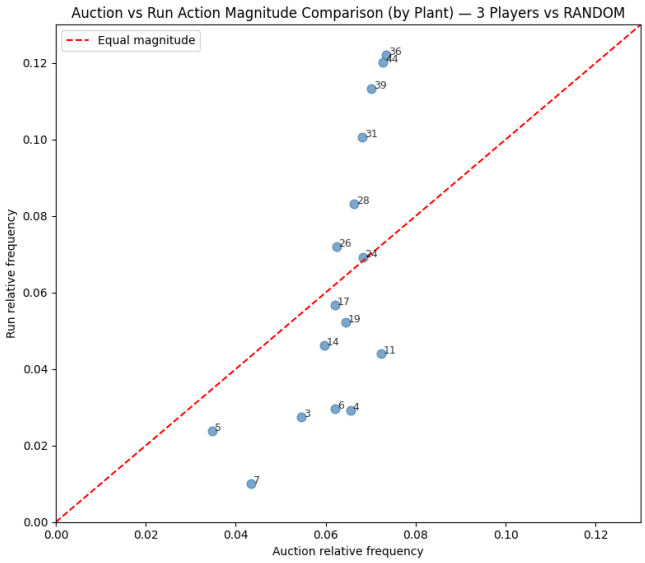
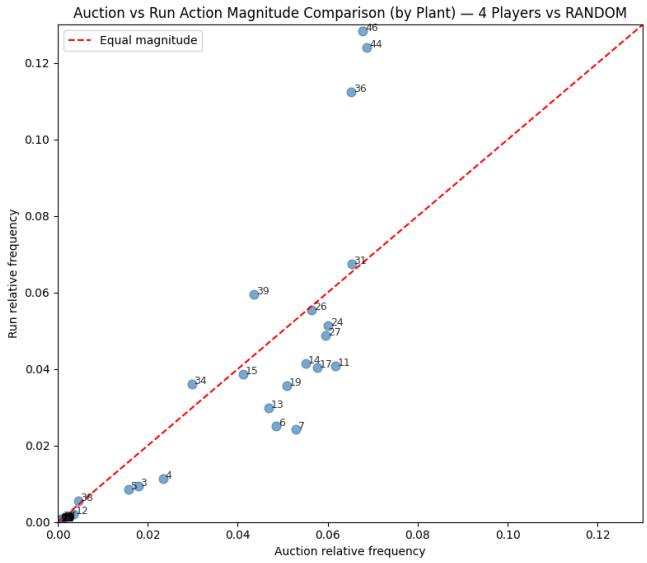


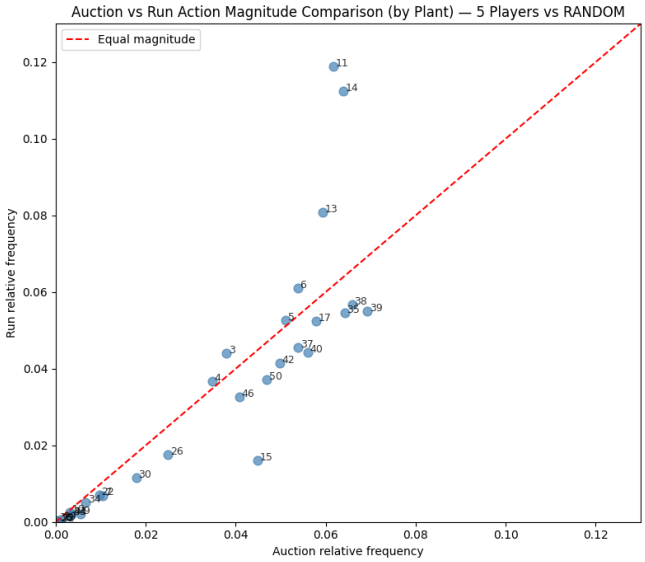
Fig. 16. City build frequency by city for the MCTS agent, aggregated across all players.



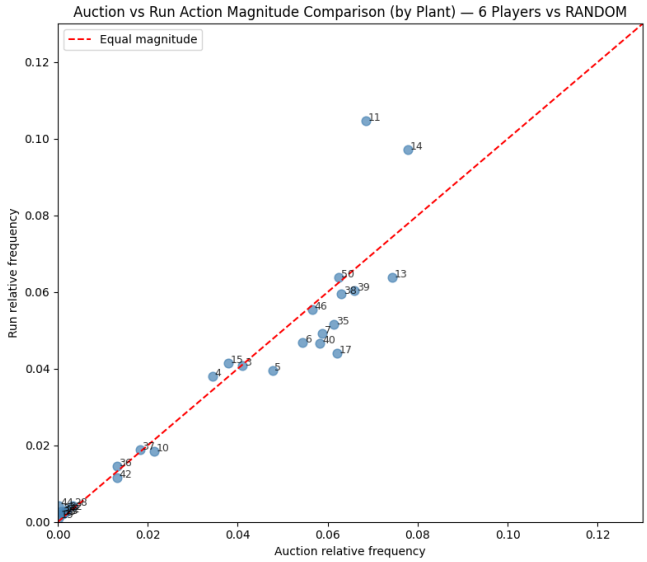
(a) 3-player



(b) 4-player

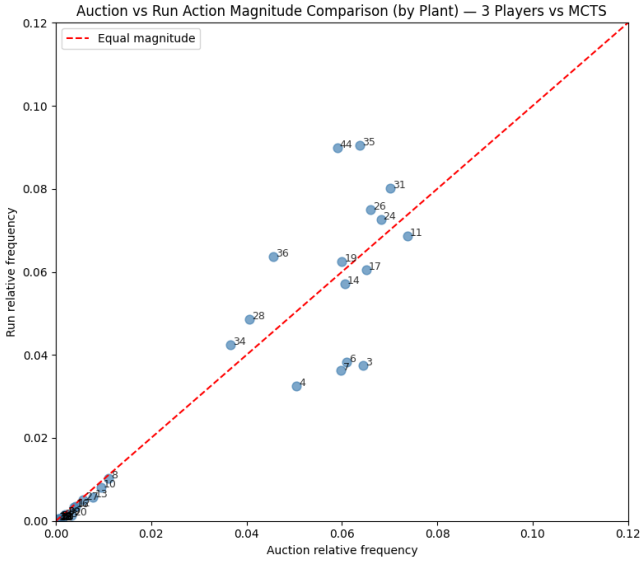


(c) 5-player

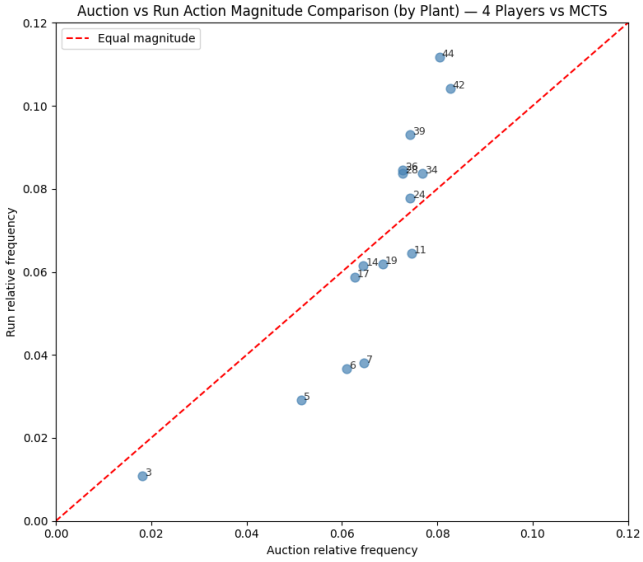


(d) 6-player

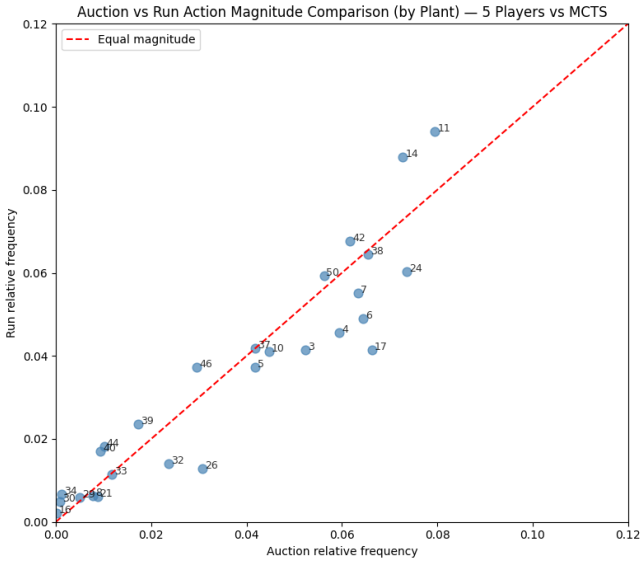
Fig. 17. Run Plant vs. Auction Plant action frequencies under the Random baseline for 3–6 player games (train/eval matched by player count).



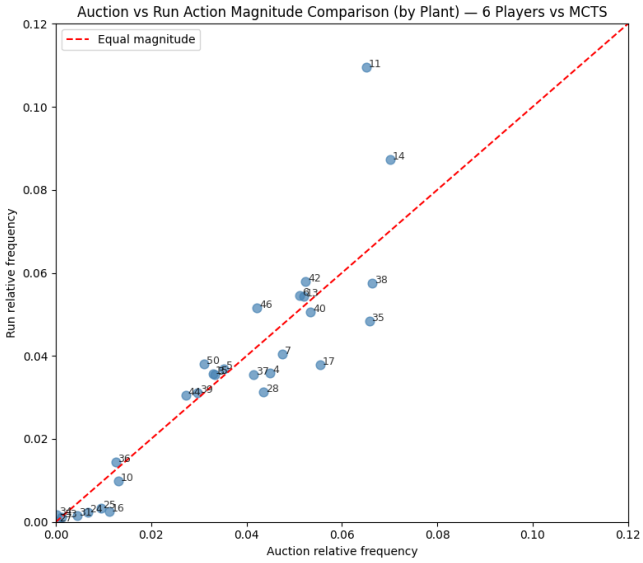
(a) 3-player



(b) 4-player



(c) 5-player



(d) 6-player

Fig. 18. Run Plant vs. Auction Plant action frequencies under MCTS opponents for 3–6 player games. Training and evaluation were performed using matching player counts.