# ONLINE WORLD-MODEL-BOOSTED PLANNING FOR MULTI-AGENT QUADRUPED SOCCER

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

## ABSTRACT

Learning coordinated behaviors in multi-agent systems remains challenging due to sparse rewards, strong inter-agent coupling, and large joint action spaces, which make model-free exploration highly sample-inefficient. We study this problem in multi-quadruped robot soccer, which is a representative and commonly-used benchmark. Specifically, each robot selects among pretrained repositioning, dribbling, and kicking skills together with continuous command parameters while competing against mobile opponents through self-play. We propose an online model-based multi-agent reinforcement learning framework in which model predictive control (MPC) method serves as a continually improving training-time teacher for decentralized high-level policies. A skill-level world model is learned online from self-play transitions and used by an MPC to optimize hybrid skill selection and continuous skill parameters. To compensate for the limited planning horizon, a learned centralized value function provides long-term guidance during action evaluation. The resulting MPC decisions are subsequently distilled into decentralized high-level policies for efficient execution. This creates a closed-loop learning process in which evolving policies generate increasingly relevant data for world-model learning, while the improving model provides progressively stronger planning targets for policy optimization. Experiments against model-free Multi-Agent Proximal Policy Optimization (MAPPO) and ablations show that the proposed method improves environment sample efficiency and overall policy performance. The project website is available at `https://anonymous.4open.science/w/skills2strategy`.

## 1 INTRODUCTION

Robot soccer provides a testbed for embodied multi-agent intelligence, linking low-level physical skills with team-level cooperation and competition. Success requires robots to translate tactical decisions into coordinated physical interactions with teammates, opponents, and the ball, making it a useful setting for studying how motor skills support collective strategies. Reinforcement learning has enabled robust low-level quadruped skills (Ji et al., 2023; Hu et al., 2024; Zhu et al., 2026), while hierarchical systems combine pretrained skills with high-level MAPPO and self-play for cooperative and competitive soccer (Su et al., 2025). However, learning effective team strategies remains interaction-intensive: discrete skill selections and continuous commands create a large joint action space, compounded by sparse rewards and coupled robot–ball interactions (Yang et al., 2024).

Existing approaches face several limitations in this setting. Model-free self-play relies heavily on environment interaction to discover effective strategies. Learned models can support action evaluation, but models trained on fixed datasets may become inaccurate as self-play changes the state-action distribution. Model-based planning also faces a horizon trade-off: short rollouts can overlook delayed rewards, whereas longer rollouts accumulate prediction errors and increase computational cost. Moreover, centralized planners rely on joint information that is unavailable to decentralized actors, limiting their direct use at deployment.

We address these challenges with an online world-model-boosted Multi-Agent Proximal Policy Optimization (MAPPO) framework. As illustrated in Figure 1, policy-generated self-play trajectories support world-model learning, Model Predictive Control (MPC) supervision, and on-policy updates.



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

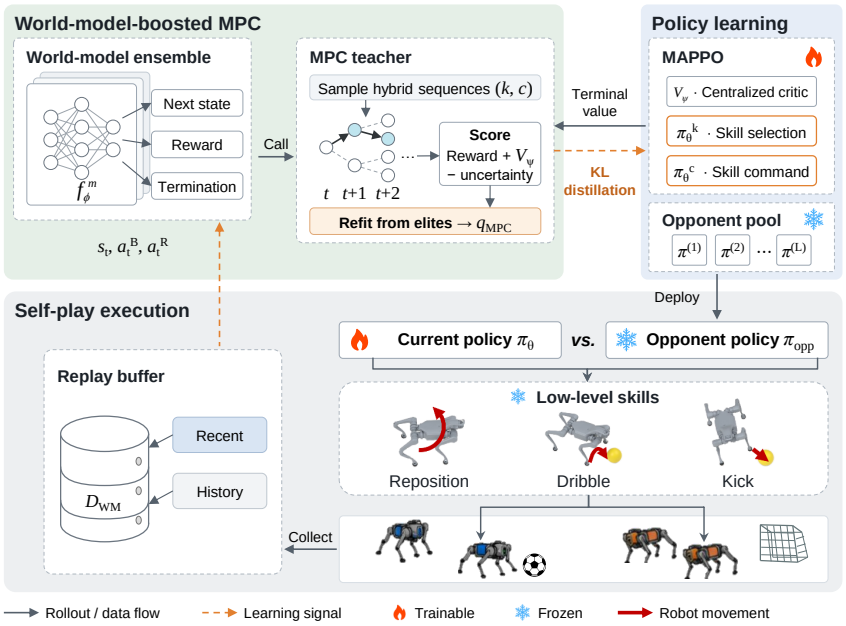



Figure 1: Overview of the proposed framework. Self-play trajectories are used to continually train the skill-level world model, which enables MPC to provide improved high-level action targets for the decentralized policies.

Historical checkpoints supply frozen opponents for both execution and planning, while the pre-trained low-level skills remain fixed.

First, a probabilistic ensemble world model predicts skill-level transitions, rewards, and termination from the global state and both teams' joint actions, separating dynamics prediction from opponent action generation. Training on recent and historical self-play data supports adaptation while retaining past experience.

Second, an uncertainty-aware MPC teacher jointly optimizes the controlled team's skill selections and continuous commands using a hybrid cross-entropy method (CEM). Candidate sequences are scored using predicted rewards, a centralized critic's terminal value, and ensemble disagreement. The terminal value estimates returns beyond short rollouts, while the disagreement penalty discourages reliance on uncertain predictions. Building on value-guided MPC (Hansen et al., 2024; Romero et al., 2026), the teacher plans over multi-agent hybrid actions with opponent responses generated by the same frozen policy used in self-play.

Third, MPC transfers privileged planning guidance into decentralized actors through auxiliary KL distillation of its optimized first-step action distributions. Actors generate all environment actions, and MAPPO returns and advantages are computed from simulator transitions. Thus, unlike methods using imagined experience for policy-gradient updates (Venugopal et al., 2024; Shi et al., 2025), our model supplies action-distribution targets for policy learning. Deployment requires only the decentralized actors and pretrained low-level skills.

Our main contributions are:

1. An online world model for multi-agent quadruped soccer with hybrid discrete–continuous actions, continually updated from self-play and historical replay to capture evolving robot–ball interaction patterns.

2. An uncertainty-aware hybrid MPC teacher combining short world-model rollouts, terminal-value guidance, and frozen-opponent policy predictions to optimize skill selections and continuous commands.

3. A world-model-boosted MAPPO framework that distills privileged MPC action distributions into decentralized actors, retaining policy-generated self-play training and requiring neither MPC nor the world model at deployment.

## 2 Related Work

Research on robot soccer spans low-level locomotion and ball manipulation, hierarchical decision-making, and multi-agent self-play. Deep reinforcement learning has enabled quadruped robots to acquire dynamic locomotion and ball-manipulation skills that are difficult to engineer manually, including robust dribbling and hierarchical ball-control behaviors (Ji et al., 2023; Zhu et al., 2026). These works primarily address physical skill execution rather than multi-robot coordination. More recent systems combine learned skills with high-level Multi-Agent Reinforcement Learning (MARL) to achieve cooperative and competitive soccer (Su et al., 2025; Tirumala et al., 2025). Large-scale league training has also produced full robotic-football strategies in simulation (Smit et al., 2023), while MQE provides environments for studying multi-agent quadruped tasks but relies mainly on a built-in locomotion controller (Xiong et al., 2024). MAPPO provides a strong centralized-training and decentralized-execution baseline, while historical self-play, PSRO, and league training expose agents to diverse and increasingly capable opponents (Yu et al., 2022; Lanctot et al., 2017; Vinyals et al., 2019). In quadruped soccer, Su et al. (2025) combine MAPPO with fictitious self-play but discretize high-level command directions to simplify learning. Such model-free, on-policy training can require substantial interaction before coordinated tactical behavior emerges (Yang et al., 2024). We instead retain continuous skill commands and use an online learned model and MPC teacher to reduce the exploration burden of high-level self-play.

Model-based reinforcement learning improves interaction efficiency by learning predictive dynamics for planning or imagined policy updates (Moerland et al., 2023). Probabilistic ensembles and CEM support uncertainty-aware trajectory optimization, while short model rollouts reduce compounding prediction error (Chua et al., 2018; Janner et al., 2019). Modern world models further demonstrate that learned dynamics can support policy learning across diverse continuous-control tasks (Hafner et al., 2025). Extending these ideas to MARL is difficult because transitions depend on coupled actions and the behavior of other agents changes during training. Existing methods address this through interaction-aware prediction, opponent modeling, structured latent dynamics, or pessimistic objectives (Krupnik et al., 2020; Wu et al., 2023; Zhang et al., 2021; Venugopal et al., 2024; Wen et al., 2026). MAMBPO, MABL, and GAWM primarily use learned models to generate imagined experience for policy optimization (Willemsen et al., 2021; Venugopal et al., 2024; Shi et al., 2025). This can improve sample efficiency, but directly training on synthetic trajectories exposes the policy to model bias, particularly as opponents and visitation distributions evolve. Our model predicts skill-level transitions conditioned on both teams' hybrid actions. It supports short-horizon planning and action-distribution supervision, while MAPPO returns and advantages are computed from simulator experience.

Learned MPC exploits a predictive model through receding-horizon optimization, but short planning horizons may lead to myopic decisions. TD-MPC and TD-MPC2 address this issue by combining short-horizon planning with a learned terminal value, and related actor–critic MPC methods use learned value estimates to provide long-term guidance (Hansen et al., 2022; 2024; Romero et al., 2026). We adopt this general principle but plan over the discrete skill selections and continuous commands of multiple robots. The centralized MAPPO critic is reused as the terminal value function, while ensemble disagreement discourages plans that rely on uncertain world-model predictions. Policy distillation and interactive imitation learning transfer teacher guidance into compact policies (Rusu et al., 2016; Ross et al., 2011). Recent methods similarly use privileged or sampling-based MPC teachers for robot learning (Amiri et al., 2026; Schuck et al., 2026). Building on these advances, our framework integrates online world-model learning, uncertainty-aware hybrid planning, and policy distillation for competitive multi-agent quadruped soccer. The model provides planning-based supervision for decentralized MAPPO actors, while policy-gradient updates use simulator experience and deployment requires neither the model nor MPC.

# 3 METHOD

## 3.1 HIERARCHICAL CONTROL AND ONLINE WORLD MODELING

We model our multi-robot soccer game between two teams, $\mathcal{B}$ and $\mathcal{R}$ as a Markov Decision Process (MDP). Each consists of $N = 2$ robots. At high-level decision step $t$, the global state is denoted by $s_t \in \mathcal{S}$, which includes the robot poses, velocities and ball state. Each robot $i$ receives a local observation $o_t^i = \mathcal{O}_i(s_t)$ including proprioception and robot identity indicators.

All robots interact with the environment through the following library of pretrained low-level skills:

$$\mathcal{K} = \{\texttt{reposition}, \texttt{dribble}, \texttt{kick}\}. \tag{1}$$

For each robot $i$ in team $X \in \{\mathcal{B}, \mathcal{R}\}$, the high-level policy outputs a hybrid action consisting of a discrete skill and its corresponding continuous command parameters:

$$a_t^{X,i} = \left(k_t^{X,i}, c_t^{X,i}\right), \qquad k_t^{X,i} \in \mathcal{K}, \quad c_t^{X,i} \in \mathbb{R}^3, \tag{2}$$

where $k_t^{X,i}$ and $c_t^{X,i}$ denote the selected skill and its associated continuous parameters, respectively. The three command components and their ranges are defined for each skill in Table 6 of Appendix B. The joint action of team $X$ at time step $t$ is given by $a_t^X = \left(a_t^{X,1}, \ldots, a_t^{X,N}\right)$, and the complete joint action of both teams is defined as $\boldsymbol{a}_t = (a_t^{\mathcal{B}}, a_t^{\mathcal{R}})$.

Executing the selected skills over one high-level interval produces the next state $s_{t+1}$ and reward $r_t$. For example, dribbling changes both the robot's position and the ball's motion, while an opponent's simultaneous movement may lead to an interception. The world model predicts these joint interaction outcomes at the skill timescale. High- and low-level reward definitions are provided in Tables 5 and Table 4 of Appendix A, respectively.

Rather than predicting complicated joint-level quadruped dynamics, the world model proposed in this work aims to predict the state transition at the high-level skill timescale.

$$p_\phi(s_{t+1}, r_t, d_t \mid s_t, a_t^{\mathcal{B}}, a_t^{\mathcal{R}}). \tag{3}$$

where $d_t$ indicates whether the episode terminates after the transition at step $t$

We implement this conditional distribution using an ensemble of $M$ probabilistic models. Each model predicts a residual distribution for the continuous state variables, together with the reward distribution and the termination probability $p_d$:

$$f_\phi^m\left(s_t, a_t^{\mathcal{B}}, a_t^{\mathcal{R}}\right) = \left(\boldsymbol{\mu}_{\Delta s}^m, \log \boldsymbol{\sigma}_{\Delta s}^{2,m}, \mu_r^m, \log \sigma_r^{2,m}, p_d^m\right). \tag{4}$$

where $m \in \{1, \ldots, M\}$ indexes the ensemble members, and $\phi$ denotes the trainable parameters of the model; $\boldsymbol{\mu}_{\Delta s}^m$ and $\mu_r^m$ are the predicted means of the state residual and reward, respectively; $\boldsymbol{\sigma}_{\Delta s}^{2,m}$ and $\sigma_r^{2,m}$ are their predicted variances; and $p_d^m$ is the predicted probability of episode termination. Then the predicted continuous state is given by

$$\hat{s}_{t+1}^m = s_t + \boldsymbol{\mu}_{\Delta s}^m. \tag{5}$$

The training objective is

$$\mathcal{L}_{\mathrm{WM}} = \frac{1}{M} \sum_{m=1}^{M} \left(\mathcal{L}_{\mathrm{state}}^m + \lambda_r \mathcal{L}_r^m + \lambda_d \mathcal{L}_d^m\right), \tag{6}$$

Here, $\mathcal{L}_{\mathrm{state}}$ denotes the negative log-likelihood of the predicted state transition, $\mathcal{L}_r$ denotes the Gaussian negative log-likelihood of the reward, and $\mathcal{L}_d$ denotes the binary cross-entropy loss for termination prediction.

The model is updated periodically throughout training using recent and historical self-play transitions. This allows it to adapt to the current policies while retaining experience from earlier opponents. We use historical self-play to reduce short-term non-stationarity. The current policy is trained against a frozen opponent sampled from a population of previous checkpoints,

$$\mathcal{P} = \left\{\pi^{(1)}, \ldots, \pi^{(L)}\right\}, \tag{7}$$

At the end of each training interval, the new policy may be added to this population and a new opponent is selected.

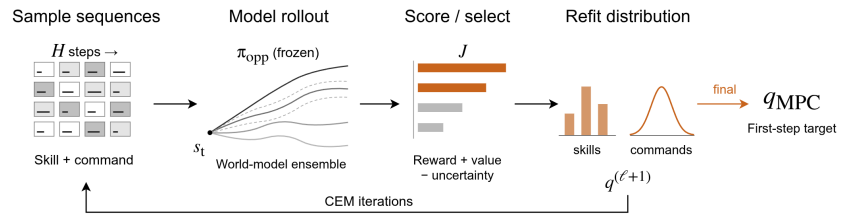



Figure 2: Overview of CEM planning within the MPC teacher. Candidate hybrid action sequences are sampled from an evolving distribution and evaluated through ensemble world-model rollouts. Elite sequences are used to refine the sampling distribution over successive CEM iterations. The optimized first-step action distribution provides a distillation target for the decentralized actors.

### 3.2 Model Predictive Control

At state $s_t$, MPC searches over the controlled-team action sequence.

$$\mathbf{A}_t^{\mathcal{B}} = \{\mathbf{a}_t^{\mathcal{B}}, \ldots, \mathbf{a}_{t+H-1}^{\mathcal{B}}\}. \tag{8}$$

At each imagined step, the opponent action is sampled from the frozen opponent policy using the opponent's predicted observation $\mathcal{O}_{\mathcal{R}}$

$$\hat{a}_{t+h}^{\mathcal{R}} \sim \pi_{\text{opp}}\left(\cdot \mid \mathcal{O}_{\mathcal{R}}(\hat{s}_{t+h})\right). \tag{9}$$

and imagined transitions are generated according to

$$\hat{s}_{t+h+1} \sim f_\phi^m\left(\hat{s}_{t+h}, \hat{\mathbf{a}}_{t+h}^{\mathcal{B}}, \hat{\mathbf{a}}_{t+h}^{\mathcal{R}}\right). \tag{10}$$

Since each action contains a discrete skill and continuous parameters, we use a hybrid CEM as shown in Figure 2. For robot $i$ and planning step $h$, CEM maintains a categorical skill distribution

$$q_{h,i}^k(k) = \Pr(k_{t+h}^{\mathcal{B},i} = k), \tag{11}$$

and a skill-conditioned Gaussian command distribution

$$c_{t+h}^{\mathcal{B},i} \mid k \sim \mathcal{N}\left(\mu_{h,i,k}, \operatorname{diag}(\sigma_{h,i,k}^2)\right). \tag{12}$$

A candidate sequence may assign dribbling to the ball-carrying robot and repositioning to its teammate, followed by a kick. The world model evaluates their joint effects under predicted opponent responses, and the highest-scoring candidates refine the CEM distributions to obtain $q_{\text{MPC}}$. For robots far from the ball, sampling is restricted to repositioning because the dribbling and kicking controllers were trained for nearby-ball interactions. This restriction applies only to teacher sampling.

To limit model error, we use a short planning horizon and bootstrap the remaining return with the centralized MAPPO critic $V_\psi$. For a candidate trajectory, the objective is

$$J = \sum_{h=0}^{H-1} \gamma^h \left(\hat{r}_{t+h} - \lambda_u U(\hat{s}_{t+h}, \hat{a}_{t+h}^{\mathcal{B}}, \hat{a}_{t+h}^{\mathcal{R}})\right) + \gamma^H V_\psi(\hat{s}_{t+H}). \tag{13}$$

where $U(\hat{s}_{t+h}, \hat{a}_{t+h}^{\mathcal{B}})$ measures the disagreement among the ensemble predictions at the corresponding imagined state and action:

$$U(\hat{s}, \hat{a}^{\mathcal{B}}, \hat{a}^{\mathcal{R}}) = \frac{1}{M}\sum_{m=1}^{M}\left(\hat{s}_{t+1}^m - \bar{s}_{t+1}\right)^\top \mathbf{W}_s \left(\hat{s}_{t+1}^m - \bar{s}_{t+1}\right), \qquad \bar{s}_{t+1} = \frac{1}{M}\sum_{j=1}^{M}\hat{s}_{t+1}^j. \tag{14}$$

where $\mathbf{W}_s$ normalizes the state dimensions before computing the disagreement. For example, dribbling may create a favorable shooting position without yielding a goal within the rollout horizon; the terminal value estimates the subsequent return. A candidate involving robot-ball contact receives an uncertainty penalty when ensemble members disagree about the resulting motion.

**Algorithm 1** Online world-model-boosted self-play training

```
Initialize decentralized actors $\pi_\theta$ and centralized critic $V_\psi$
Initialize world-model ensemble $\{f_\phi^m\}_{m=1}^M$ , replay buffer $\mathcal{D}_{\mathrm{WM}}$ and opponent population $\mathcal{P}$
for each training interval do
    Sample a frozen opponent $\pi_{\mathrm{opp}}$ from $\mathcal{P}$
    Collect on-policy self-play trajectories using $\pi_\theta$ and store the transitions in $\mathcal{D}_{\mathrm{WM}}$
    if in warm-up period then
        Update $\pi_\theta$ and $V_\psi$ using the MAPPO objective
    else
        Select training states from the collected trajectories
        Run hybrid CEM-MPC from each selected state
        Compute $\mathcal{L}_{\mathrm{MPC}}$ and update $\pi_\theta$ using Eq. (20)
    end if
    Update the world-model ensemble using minibatches from $\mathcal{D}_{\mathrm{WM}}$ and $V_\psi$
    Add the current policy checkpoint to $\mathcal{P}$
end for
```

### 3.3 MPC-Guided MAPPO and Online Training

Environment actions are sampled from the current policy, preserving the on-policy MAPPO objective. For robot $i$, the PPO ratio is

$$\rho_t^i(\theta) = \frac{\pi_\theta(a_t^i \mid o_t^i)}{\pi_{\theta^{\mathrm{old}}}(a_t^i \mid o_t^i)}, \tag{15}$$

with clipped objective

$$\mathcal{J}_{\mathrm{PPO}}^i = \mathbb{E}_t \left[ \min \left( \rho_t^i A_t, \mathrm{clip}(\rho_t^i, 1-\epsilon, 1+\epsilon) A_t \right) \right]. \tag{16}$$

MPC is used as an auxiliary teacher rather than as the behavior policy. For selected states, its skill and command distributions are distilled into the decentralized actors:

$$\mathcal{L}_{\mathrm{skill}}^i = D_{\mathrm{KL}} \left( q_{\mathrm{MPC},i}^k \| \pi_\theta^k(\cdot \mid o_t^i) \right), \tag{17}$$

$$\mathcal{L}_{\mathrm{cmd}}^i = \sum_{k \in \mathcal{K}} q_{\mathrm{MPC},i}^k(k) D_{\mathrm{KL}} \left( q_{\mathrm{MPC},i}^c(\cdot \mid k) \| \pi_\theta^c(\cdot \mid o_t^i, k) \right). \tag{18}$$

where the resulting CEM distributions are denoted by $q_{\mathrm{MPC},i}^k$ for the skill distribution and $q_{\mathrm{MPC},i}^c(\cdot \mid k)$ for the skill-conditioned command distribution.

The total distillation loss is

$$\mathcal{L}_{\mathrm{MPC}} = \frac{1}{N} \sum_{i=1}^N \left( \mathcal{L}_{\mathrm{skill}}^i + \lambda_c \mathcal{L}_{\mathrm{cmd}}^i \right), \tag{19}$$

and the actor objective is

$$\mathcal{L}_{\mathrm{actor}} = -\mathcal{J}_{\mathrm{PPO}} + \beta_{\mathrm{MPC}} \mathcal{L}_{\mathrm{MPC}} - \lambda_{\mathrm{ent}} \mathcal{H}(\pi_\theta). \tag{20}$$

At the beginning of training, the policy uses standard MAPPO self-play while collecting transitions for $\mathcal{D}_{\mathrm{WM}}$. After a warm-up period of $T_{\mathrm{warm}}$ environment steps, MPC guidance is enabled, optionally with a gradual increase in its coefficient:

$$\beta_{\mathrm{MPC}}(n) = \beta_{\max} g(n), \tag{21}$$

where $g(n)$ increases from zero to one after warm-up. The complete training procedure is summarized in Algorithm 1.

## 4 Experiments

The experiments are conducted in Isaac Gym (Makoviychuk et al., 2021) using the Unitree As2 robot model. Each team contains $N = 2$ robots. The low-level repositioning, dribbling, and kicking

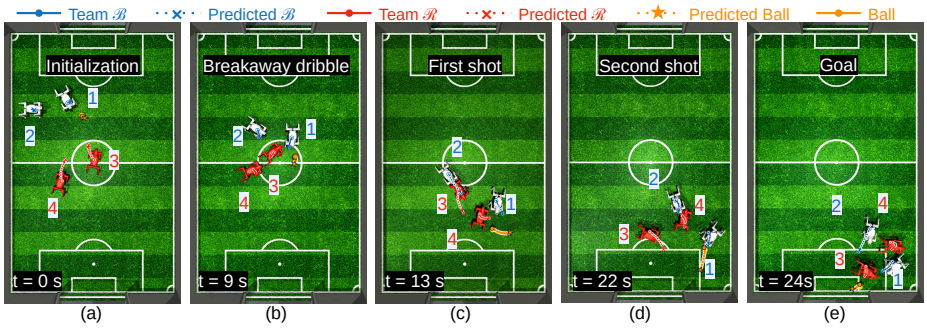



Figure 3: An attacking sequence with simulator snapshots and corresponding MPC-predicted trajectories: (a) initial configuration; (b) robot 1 dribbles past a robot 3; (c) its first shot gets blocked by robot 4, robot 2 repositioning but gets blocked by robot 3; (d) robot 1 attempts a second shot; (e) the ball enters the goal.

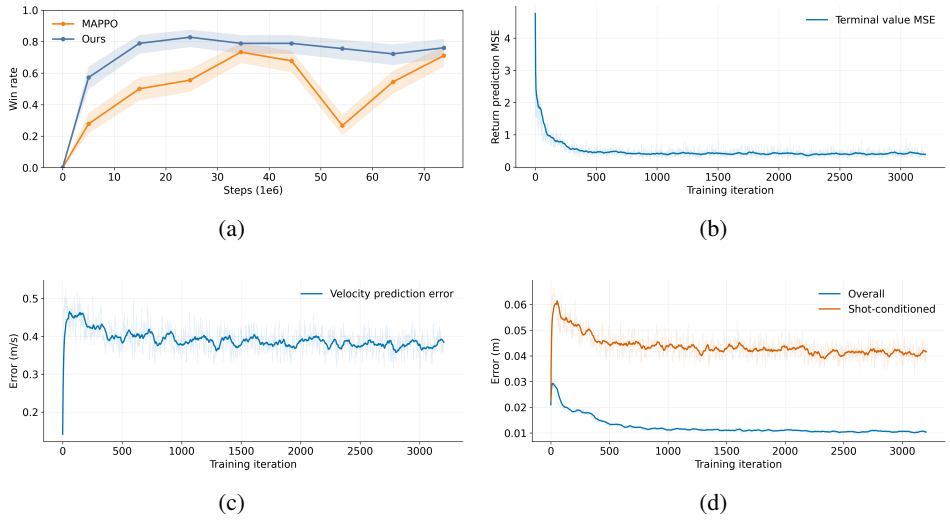



Figure 4: Training dynamics of the proposed method. (a) Win rate of policy checkpoints evaluated against a fixed pool of opponent checkpoints. (b) Terminal-value prediction error. (c) Velocity prediction error. (d) Ball-position prediction error, reported for all transitions and for shot-conditioned transitions separately.

controllers are pretrained before high-level training and remain frozen throughout all experiments. The high-level policy operates at a fixed decision interval, and each selected skill is executed for the corresponding number of low-level simulation steps.

We evaluate four aspects of the framework: 1) environment-sample efficiency and head-to-head performance; 2) prediction quality and computational cost across rollout horizons; 3) the effects of online world-model updates and historical replay; and 4) the contributions of terminal-value guidance and uncertainty penalization.

### 4.1 Main Results

In Figure 3, we illustrate an attacking sequence involving dribbling, a blocked shot, and a successful second attempt. The MPC trajectories are computed retrospectively from recorded states for visualization and do not control the displayed rollout.

Figure 4 (a) compares win-rate learning curves under the same fixed opponent pool. Our method shows faster early learning and a higher final win rate than MAPPO under the same fixed evaluation

pool. The narrowing gap later in training suggests that additional model-free interaction partially closes the difference, whereas the early improvement is consistent with the benefit of world-model-boosted policy learning. Figures 4 (b)-(d) show the corresponding world-model errors. As the policies learn basic coordination, the self-play data become more strongly coupled and harder to predict. Online updates then reduce the position-prediction error, while the velocity error remains higher because kicking causes abrupt ball-motion changes.

We further compare the methods in head-to-head matches, as reported in Table 1. For each matchup, we conduct 100 matches per seed across 10 random seeds and report the mean win rate and its variability across seeds. Draws were not taken into account. In these matches, our method wins against MAPPO and discrete MAPPO with rates of $59.47\%$ and $93.72\%$, respectively, while MAPPO wins against discrete MAPPO at $92.43\%$. These results indicate a moderate advantage over continuous-action MAPPO and a much larger gap between the continuous- and discrete-action formulations.

Discrete MAPPO uses eight fixed-direction commands with a fixed velocity magnitude. Although this removes the continuous command variables, it restricts the available direction and speed choices, which can limit precise coordination. We implemented this baseline based on Su et al. (2025); its results may therefore differ from the original work.

Table 1: Win rate against different opponents

| Evaluated match | | | Win rate |
|---|---|---|---|
| Ours | v.s. | MAPPO | $(59.47 \pm 0.92)\%$ |
| Ours | v.s. | Discrete MAPPO | $(93.72 \pm 0.54)\%$ |
| MAPPO | v.s. | Discrete MAPPO | $(92.43 \pm 0.56\%)$ |

### 4.2 Planning Horizon and World-Model Quality

We next examine whether the learned world model and terminal value function provide a reliable basis for short-horizon planning. Table 2 reports position-prediction error, ensemble uncertainty, terminal value MSE, and rollout time for different planning horizons. The position error contains the error for ball and robots unlike Figure 4 (d). Longer rollouts incur greater prediction error and computational cost. However, optimizing only rewards within a short horizon can favor actions with immediate benefits over those whose rewards occur later.

The terminal value term addresses this limitation by estimating the remaining return beyond the model rollout. Consequently, the planner can compare candidate sequences using both their short-term rewards and the estimated long-term value of their endpoint states, without requiring a very large planning horizon (Chen & Allgöwer, 1998). The terminal value does not guarantee globally optimal planning; instead, its effectiveness depends on the accuracy of the learned critic and the world-model rollout.

Table 2: World-model accuracy across planning horizons.

| Horizon | Position Error (cm) | Uncertainty | Terminal value MSE | Rollout Time (ms/batch) |
|---|---|---|---|---|
| 1 | 4.1 | 1.3 | 0.71 | 10.8 |
| 2 | 8.7 | 2.5 | 0.89 | 21.4 |
| 4 | 11.6 | 4.6 | 1.16 | 41.9 |
| 10 | 17.6 | 15.3 | 1.68 | 105.8 |

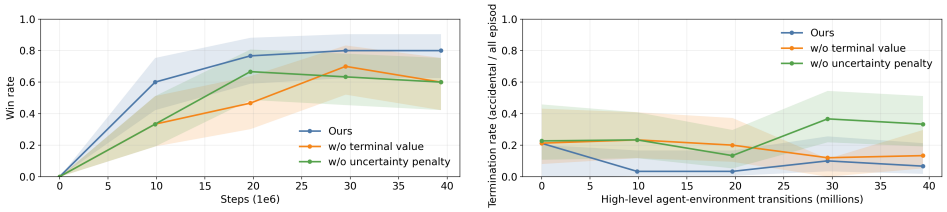
A key design choice is to train the world model continuously together with the evolving self-play policy. Table 3 compares frozen, recent-online, our method. Our method, which combines online training and replay buffer, achieves the lowest errors. This result indicates that recent self-play data is necessary to track the evolving opponent distribution, while historical replay preserves coverage of earlier opponent behaviors.

Table 3: World model training methods comparison

| World Model | Position Error (cm) | Uncertainty |
|---|---|---|
| Frozen | 14.0 | 10.8 |
| Online | 18.1 | 12.9 |
| **Ours** | **11.6** | **4.6** |

### 4.3 Ablation Study

We investigate the effects of the terminal-value term and the uncertainty penalty in the MPC objective. We compare the full method with two variants: one without the terminal-value term and one without the uncertainty penalty.




(a) (b)

Figure 5: Ablation of terminal-value guidance and uncertainty penalization. (a) Win rate versus training transitions. (b) Accidental-termination rate, defined as the fraction of evaluation episodes ending in an accidental termination.

Figure 5 compares the learning curves and termination rates of the three variants. Removing the terminal-value term leads to slower early improvement and lower final win rate. This result is consistent with the role of the terminal value in providing an estimate of the return beyond the finite model rollout, thereby reducing the myopic bias of short-horizon reward optimization. In contrast, the full method reaches a higher win rate earlier and maintains the best performance throughout the later stages of training.

Removing the uncertainty penalty produces a larger termination rate as shown in Figure 5, particularly after the policy begins to acquire more effective interaction behaviors. This suggests that, without uncertainty regularization, the MPC teacher is more likely to select candidate sequences that rely on poorly modeled state-action regions. The uncertainty penalty suppresses such model exploitation and results in more stable training. The full method combines both effects: the terminal value provides long-term guidance for short-horizon planning, while the uncertainty penalty improves the reliability of model-based action evaluation.

## 5 Conclusion

We presented an online world-model-boosted framework for multi-agent quadruped soccer with hybrid discrete-continuous actions. A probabilistic ensemble world model is continually updated using self-play transitions and historical replay to capture interactions among robots and the ball. An uncertainty-aware hybrid MPC teacher combines short model rollouts, terminal-value guidance, and frozen-opponent policy predictions to optimize skill selections and continuous commands. Its action distributions provide auxiliary KL supervision for MAPPO actors, while environment actions remain policy-generated. Deployment requires only the decentralized actors and frozen low-level skills.

Experiments show faster early learning and a modest advantage over MAPPO. World-model comparisons support combining online updates with historical replay, while ablations show that both terminal-value guidance and uncertainty penalization contribute to higher final win rates within the evaluated training budget. Future work will investigate more accurate interaction models and transfer to physical quadruped teams.

## 6 REPRODUCIBILITY STATEMENT

We are committed to ensuring the reproducibility of all results presented in this paper. To facilitate reproduction, we publicly release all source code, including those to train the high-level policy, low-level skills, and world model, at our project page:

`https://anonymous.4open.science/w/skills2strategy`.

## 7 LLM USAGE

LLMs were not used in the conceptualization of the study, formulation of its key contributions, or identification of relevant prior work. They were used for assistance with code debugging and manuscript organization and revision. The authors reviewed all suggestions and take responsibility for the scientific claims, experimental results, and final manuscript.

## REFERENCES

Mohsen Amiri, Mohsen Amiri, Ali Beikmohammadi, Sindri Magnússon, and Mehdi Hosseinzadeh. PriPG-RL: Privileged planner-guided reinforcement learning for partially observable systems with anytime-feasible MPC, 2026.

Hong Chen and Frank Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, pp. 4759–4770, 2018.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

Nicklas A. Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8387–8406. PMLR, 2022.

Yutong Hu, Kehan Wen, and Fisher Yu. DexDribbler: Learning dexterous soccer manipulation via dynamic supervision. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12910–12917, 2024.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 32, pp. 12498–12509, 2019.

Yandong Ji, Gabriel B. Margolis, and Pulkit Agrawal. DribbleBot: Dynamic legged manipulation in the wild. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5155–5162, 2023.

Orr Krupnik, Igor Mordatch, and Aviv Tamar. Multi-agent reinforcement learning with multi-step generative models. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 776–790. PMLR, 2020.

Marc Lanctot, Vinícius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4190–4203, 2017.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac Gym: High performance GPU-based physics simulation for robot learning, 2021.

Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based reinforcement learning: A survey. *Foundations and Trends in Machine Learning*, 16(1):1–118, 2023.

Ángel Romero, Elie Aljalbout, Yunlong Song, and Davide Scaramuzza. Actor–critic model predictive control: Differentiable optimization meets reinforcement learning for agile flight. *IEEE Transactions on Robotics*, 42:673–692, 2026.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011.

Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. In *International Conference on Learning Representations*, 2016.

Martin Schuck, Maks Sorokin, Simone Manni, Duy Ta, Angela P. Schoellig, Marco Hutter, Simon Le Cleac'h, and Jan Brüdigam. Learning loco-manipulation from SMPC demonstrations with sparse offline-to-online RL, 2026.

Zifeng Shi, Meiqin Liu, Senlin Zhang, Ronghao Zheng, Shanling Dong, and Ping Wei. GAWM: Global-aware world model for multi-agent reinforcement learning, 2025.

Andries Smit, Herman A. Engelbrecht, Willie Brink, and Arnu Pretorius. Scaling multi-agent reinforcement learning to full 11 versus 11 simulated robotic football. *Autonomous Agents and Multi-Agent Systems*, 37(1):20, 2023.

Zhi Su, Yuman Gao, Emily Lukas, Yunfei Li, Jiaze Cai, Faris Tulbah, Fei Gao, Chao Yu, Zhongyu Li, Yi Wu, and Koushil Sreenath. Toward real-world cooperative and competitive soccer with quadrupedal robot teams. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4723–4742. PMLR, 2025.

Dhruva Tirumala, Markus Wulfmeier, Ben Moran, Sandy Huang, Jan Humplik, Guy Lever, Tuomas Haarnoja, Leonard Hasenclever, Arunkumar Byravan, Nathan Batchelor, Neil Sreendra, Kushal Patel, Marlon Gwira, Francesco Nori, Martin Riedmiller, and Nicolas Heess. Learning robot soccer from egocentric vision with deep reinforcement learning. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 165–184. PMLR, 2025.

Aravind Venugopal, Stephanie Milani, Fei Fang, and Balaraman Ravindran. MABL: Bi-level latent-variable world model for sample-efficient multi-agent reinforcement learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1865–1873, 2024.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354, 2019.

Ruoqi Wen, Rongpeng Li, Xing Xu, Mahdi Boloursaz Mashhadi, Pei Xiao, and Zhifeng Zhao. Multi-agent uncertainty-aware pessimistic model-based reinforcement learning for connected autonomous vehicles. *IEEE Transactions on Mobile Computing*, pp. 1–18, 2026.

Daniël Willemsen, Mario Coppola, and Guido C. H. E. de Croon. MAMBPO: Sample-efficient multi-robot reinforcement learning using learned world models. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5635–5640. IEEE, 2021.

Zifan Wu, Chao Yu, Chen Chen, Jianye Hao, and Hankz Hankui Zhuo. Models as agents: Optimizing multi-step predictions of interactive local models in model-based multi-agent reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):10435–10443, 2023.

Ziyan Xiong, Bo Chen, Shiyu Huang, Wei-Wei Tu, Zhaofeng He, and Yang Gao. Mqe: Unleashing the power of interaction with multi-agent quadruped environment, 2024.

Yaodong Yang, Guangyong Chen, Jianye Hao, and Pheng-Ann Heng. Sample-efficient multiagent reinforcement learning with reset replay. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 55961–55975. PMLR, 2024.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24611–24624, 2022.

Weinan Zhang, Xihuai Wang, Jian Shen, and Ming Zhou. Model-based multi-agent policy optimization with adaptive opponent-wise rollouts. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 3384–3391, 2021.

Dongjie Zhu, Zhuo Yang, Xuesong Li, Wenjun Xu, Qi Liu, and Xiang Li. Dynamic legged ball manipulation on rugged terrains with hierarchical reinforcement learning. *IEEE Robotics and Automation Letters*, 11(6):6815–6822, 2026.

# A APPENDIX REWARD SETTING

Table 4: Low level reward setting

| Skill | Reward term | Brief explanation | Coefficient |
|---|---|---|---|
| Common | orientation | Penalize body roll and pitch. | $-5.0$ |
| | torques | Penalize squared joint torques. | $-3 \times 10^{-5}$ |
| | dof_vel | Penalize squared joint velocities. | $-10^{-4}$ |
| | dof_acc | Penalize squared joint accelerations. | $-2.5 \times 10^{-7}$ |
| | collision | Penalize contacts on selected robot bodies. | $-1.0$ |
| | action_rate | Penalize changes between consecutive actions. | $-0.01$ |
| | contacts_shaped_force | Penalize foot contact forces during swing. | 4.0 |
| | contacts_shaped_vel | Penalize foot motion during stance. | 4.0 |
| | dof_pos_limits | Penalize joint position limit violations. | $-10.0$ |
| | dof_pos | Penalize deviations from default joint positions. | $-0.05$ |
| | action_smoothness_1 | Penalize first differences of joint position targets. | $-0.1$ |
| | action_smoothness_2 | Penalize second differences of joint position targets. | $-0.1$ |
| Reposition | walking_vel | Track commanded planar velocity and yaw rate. | 4.0 |
| Dribble | robot_ball_vel | Encourage approaching the ball at the commanded speed. | 0.5 |
| | robot_ball_pos | Keep the ball near the desired position ahead of the robot. | 4.0 |
| | backward_motion | Penalize robot motion opposite the requested direction. | 2.0 |
| | ball_vel | Track the commanded planar ball velocity. | 4.0 |
| | robot_ball_yaw | Keep the ball aligned with the robot's forward direction. | 4.0 |
| | ball_vel_norm | Track the commanded ball speed. | 4.0 |
| | ball_vel_angle | Align ball motion with the commanded direction. | 4.0 |
| | tracking_ang_vel | Track the commanded robot yaw rate. | 1.0 |
| Kick | ball_vel | Track target ball velocity after the kick and separation. | 5.0 |
| | ball_forward | Reward ball motion along the commanded shot direction. | 8.0 |
| | ball_vel_norm | Track target ball speed after the kick and separation. | 1.5 |
| | ball_vel_angle | Align post-kick ball motion with the shot direction. | 1.5 |
| | ball_out | Reward ball travel and speed along the shot direction. | 7.0 |
| | robot_approach_ball | Reward signed progress toward the pre-kick setup position. | 1.5 |
| | excess_yaw | Penalize excessive yaw rate before the kick. | $-0.1$ |
| | launch* | Reward a qualifying ball launch. | 5.0 |
| | success* | Reward meeting shot speed, alignment, and travel criteria. | 25.0 |
| | failure* | Penalize a failed shooting attempt. | $-2.0$ |

*Starred terms are event rewards; other terms are timestep-scaled.

Table 5: High level reward setting

| **Reward term** | **Brief explanation** | **Coefficient** |
|---|---|---|
| `goal`* | Score in the opponent's goal. | 75.0 |
| `opponent_goal`* | Concede a goal. | −75.0 |
| `accidental_termination`* | Learning-team failure causing termination. | −5.0 |
| `ball_goal_progress` | Signed ball progress toward the goal. | 2.0 |
| `goalward_ball_velocity` | Ball velocity directed toward the goal. | 0.75 |
| `approach_ball` | Attacker approaches the ball from outside skill range. | 1.0 |
| `dribble_ball_control` | Controlled, goalward dribbling. | 1.0 |
| `shoot_launch`* | Valid shot launches the ball toward the goal. | 5.0 |
| `invalid_skill` | Penalize invalid skill requests. | −2.0 |
| `robot_collision` | Penalize unsafe clearance between robots. | −2.0 |
| `support_lane` | Penalize support robots obstructing the shot lane. | −0.5 |
| `attack_position` | Attacker progresses toward a position behind a slow ball. | 1.0 |

*Starred terms are event rewards; other terms are timestep-scaled.

## B CONTINUOUS COMMAND SETTING

Table 6: Low-level skills and their continuous command spaces.

| Low-level skill | Continuous output definition | Range |
|---|---|---|
| Reposition | Robot planar velocity $(v_x, v_y)$<br>Robot yaw rate $\omega_z$ | $v_x, v_y \in [-1.5, 1.5]\,\mathrm{m/s}$<br>$\omega_z \in [-1, 1]\,\mathrm{rad/s}$ |
| Dribble | Target ball velocity $(v_x^b, v_y^b)$<br>Robot yaw rate $\omega_z$ | $v_x^b, v_y^b \in [-1.5, 1.5]\,\mathrm{m/s}$<br>$\omega_z \in [-1, 1]\,\mathrm{rad/s}$ |
| Kick | Target ball velocity $(v_x^b, v_y^b, _)$ | $v_x^b, v_y^b \in [-3, 3]\,\mathrm{m/s}$ |

*The third command component is unused for kicking.