Social World Model-Augmented Mechanism Design Policy Learning

Xiaoyuan Zhang^{1,2,3*} **Yizhe Huang**^{1,2*} **Chengdong Ma**^{1,3} **Zhixun Chen**^{4} **Long Ma**^{5}
Yali Du^{6} **Song-Chun Zhu**^{2,1,3†} **Yaodong Yang**^{1,3†} **Xue Feng**^{2†}
[1]Institute for Artificial Intelligence, Peking University
[2]State Key Laboratory of General Artificial Intelligence, BIGAI, Beijing, China
[3] State Key Laboratory of General Artificial Intelligence, Peking University, Beijing, China
[4]The Hong Kong University of Science and Technology (Guangzhou)
[5]Center for Data Science, Academy for Advanced Interdisciplinary Studies, Peking University
[6]King's College London

Abstract

Designing adaptive mechanisms to align individual and collective interests remains a central challenge in artificial social intelligence. Existing methods often struggle with modeling heterogeneous agents possessing persistent latent traits (e.g., skills, preferences) and dealing with complex multi-agent system dynamics. These challenges are compounded by the critical need for high sample efficiency due to costly real-world interactions. World Models, by learning to predict environmental dynamics, offer a promising pathway to enhance mechanism design in heterogeneous and complex systems. In this paper, we introduce a novel method named **SWM-AP** (**S**ocial **W**orld **M**odel-**A**ugmented Mechanism Design **P**olicy Learning), which learns a social world model hierarchically modeling agents' behavior to enhance mechanism design. Specifically, the social world model infers agents' traits from their interaction trajectories and learns a trait-based model to predict agents' responses to the deployed mechanisms. The mechanism design policy collects extensive training trajectories by interacting with the social world model, while concurrently inferring agents' traits online during real-world interactions to further boost policy learning efficiency. Experiments in diverse settings (tax policy design, team coordination, and facility location) demonstrate that SWM-AP outperforms established model-based and model-free RL baselines in cumulative rewards and sample efficiency.

1 Introduction

Mechanism design, the art of engineering incentive structures to guide self-interested agents towards desirable collective outcomes, underpins a vast array of societal functions, from resource allocation in digital economies and smart cities to the formulation of public policies [23]. Its profound significance lies in its potential to maximize social welfare, foster efficient cooperation, and resolve complex coordination problems in multi-agent systems [25]. However, traditional mechanism design paradigms often grapple with fundamental challenges inherent in real-world social systems. Chief among these is agent heterogeneity. Real-world populations consist of diverse individuals possessing persistent yet often unobservable latent traits (e.g., skills, preferences, risk attitudes), which critically influence their responses to incentives [21, 1]. Classical models frequently resort to simplifying assumptions of homogeneity or rely on unrealistic full information, leading to suboptimal or ineffective mechanisms.

[*] Equal contribution

[†] Equal corresponding authors. Project website: https://sites.google.com/view/swm-ap/

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

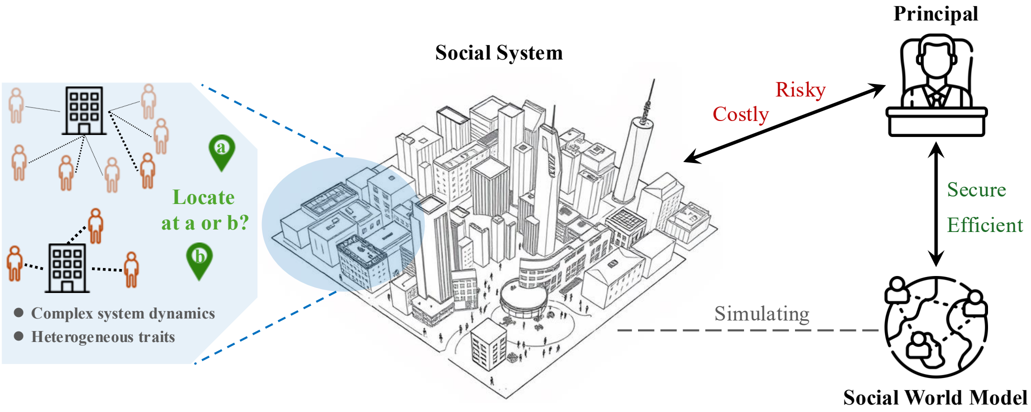


Figure 1: The AI mechanism designer (*principal*) makes decisions within a complex social system, exemplified by the facility location problem(see subsection 4.1 for details) shown on the left of the figure, which involves selecting the optimal site from potential locations (a or b). Directly interacting with the real social system is costly and risky. Such social systems typically present two major challenges. (1) Complex system dynamics, where individual interactions and environmental changes give rise to intricate and evolving system behaviors. (2) Heterogeneous agent traits, where individuals possess diverse latent preferences and needs (as illustrated in the magnified area on the left, where color intensity indicates preference for a facility, and the overall agent distribution also impacts the optimal location choice). Our proposed Social World Model aids the *principal* by simulating interaction in a secure and efficient manner.

Furthermore, these systems are characterized by complex system dynamics, which are difficult to capture using static, equilibrium-based analyses. Compounding these issues is the pervasive information asymmetry, where mechanism designers typically lack direct access to the crucial latent traits driving agent behavior. The advent of Artificial Intelligence, particularly Reinforcement Learning (RL), has ushered in a new era for mechanism design [39], offering unprecedented capabilities to develop adaptive and data-driven mechanisms capable of navigating increasingly complex and dynamic environments.

Model-Based Reinforcement Learning (MBRL) has emerged as a promising avenue for enhancing the efficacy and sample efficiency of mechanism design. By learning a model of the environment's dynamics, MBRL allows simulating trial-and-error exploration and counterfactual reasoning [22, 8], significantly reducing the reliance on costly real-world interactions, a critical advantage in high-stakes social systems. Despite this potential, the direct application of existing MBRL techniques to social mechanism design faces considerable hurdles. A primary limitation is the persistent neglect of agent heterogeneity. Many contemporary world models still treat agents as homogeneous entities or struggle to effectively represent and leverage their distinguishing latent traits. This oversight directly conflicts with the core need to design mechanisms tailored to the characteristics of the diverse agents [26, 12]. Moreover, modeling the intricate complexity of social interactions, encompassing cooperation, competition, and influence dynamics that can lead to highly non-linear and emergent system behaviors, poses a substantial challenge for standard world models [34], especially when individual agents are fundamentally driven by their underlying, unobservable traits.

To address these pressing challenges, we introduce a novel framework, named Social World Model-Augmented Mechanism Design Policy Learning (SWM-AP). Our approach, leveraging a model-based reinforcement learning paradigm, comprises two primary, interconnected components. The first is a sophisticated **Social World Model** (SWM), engineered to perform latent trait inference by unearthing agents' persistent hidden characteristics (e.g., skills, preferences) from their interaction trajectories in an unsupervised manner. It also learns trait-conditioned system dynamics, predicting how the social system evolves (i.e., state transitions and reward generation) as a function of these inferred traits and deployed mechanisms. The second core component is the **Mechanism Design Policy**. It is responsible for deploying optimal incentive mechanisms. This policy leverages the capabilities of SWM in two key ways: first, its prior mind tracker module conducts real-time inference of background agents' traits using the posterior mind tracker of SWM as the supervision signal;

second, it interacts with SWM's simulative environment to efficiently explore and refine its strategies. This synergy allows the Mechanism Design Policy to devise more adaptive, targeted, and ultimately effective incentive structures, aiming to maximize social welfare while minimizing the need for costly real-world samples, as shown in Figure 1.

The paper is structured as follows. We first review relevant literature on world models and mechanism design. Subsequently, we detail our SWM-AP framework with a theoretical analysis of algorithm feasibility. Through extensive numerical experiments across multiple scenarios, including facility location games, team optimization, and tax policy design, we validate our method's effectiveness. We conclude by summarizing contributions and proposing future directions for applying world models to real-world mechanism design research.

2 Related Works

2.1 Mechanism Design in Reinforcement Learning

The integration of Reinforcement Learning (RL) with mechanism design offers a powerful paradigm for dynamic systems, overcoming limitations of classical game-theoretic approaches tied to static equilibria and strict rationality. While foundational theories like Mirrlees' theory of optimal taxation [21] exist, they often struggle in dynamic, heterogeneous environments where agent preferences and capabilities evolve [4]. RL enables data-driven policy optimization via sequential interaction in complex settings [32, 18, 11, 2, 27]. However, contemporary RL-based mechanism design often oversimplifies agents as homogeneous, neglecting crucial cognitive traits (e.g., risk tolerance) that drive real-world decisions [30, 3]. While Inverse Reinforcement Learning (IRL) methods like 'Democratic AI' can infer latent preferences, they face scalability issues in co-training [16, 17]. Cognitive-aware RL advances this frontier: The M^{3}RL framework [31] incorporates psychological states into policy adaptation but relies on explicit reward structures. Our work bridges these gaps by introducing a social world model that co-optimizes mechanism design. Many MARL approaches flat social structures for cooperative coordination [33, 24, 35, 15], while our hierarchical mechanism design guides self-interested agents.

2.2 World Model

Model-Based Reinforcement Learning methods learn dynamic models to guide policy optimization, reducing sample complexity while maintaining performance. The learned dynamic models fall into two categories. First is enhancing model-free methods with the learned model. Model-enhanced methods include MBPO [14], which trains the SAC or PPO algorithm using generated and real trajectories. Similar ideas have been extended to offline model-based RL settings [36]. Impressive advancements have also been made in learning dynamic changes in latent variable spaces [6, 5, 7, 8]. Furthermore, the application of transformers as world models [20] has demonstrated robust performance in humanoid robots [28, 38]. The second way is to use the model for planning. TD-MPC [10, 9] incorporates terminal value estimates for long-term reward estimates. Some existing works on multi-agent reinforcement learning employ world models to simulate the dynamics of multiple systems [37, 34, 19]. However, due to their lack of modeling complex social relationships among agents or explicit specification of agents' inherent attributes to simplify the problem, these approaches face challenges in deployment to real-world social environments. In contrast, our methodology simulates the dynamics of multi-agent systems characterized by individual heterogeneity and complex interaction relationships.

3 Method

In this paper, we propose a Social World Model-Augmented Mechanism Design Policy Learning **(SWM-AP)** approach, as illustrated in Figure 2 and Algorithm 1. In Subsection 3.1, we formally define the research problem through decision-making processes. Subsection 3.2 details the learning architecture of our SWM-AP approach, including the specifics of the Social World Model with its hidden trait inference tracker, and the training of the Mechanism Design Policy. Subsection 3.3 provides theoretical justification via ELBO derivation for our approach combining latent trait inference with world model learning, demonstrating the feasibility of jointly learning latent traits and system dynamics, thereby supporting our approach of using trait inference to enhance the state prediction accuracy.

3.1 Problem Formulation

We formalize AI-driven mechanism design as an episodic Markov game between an institutional planner (*principal*) and a population of background agents. The *principal* operates as an algorithmic policy designer that dynamically deploys incentive mechanisms to optimize social welfare(the sum of all the background agents' returns). Each background agent maintains a fixed trait (such as skill or preference), which regulates the agent's response to the mechanism and shapes its behavior during interactions with other agents. These traits are private and unobservable to the *principal*, constituting a central challenge for adaptive mechanism design.

This problem can be succinctly summarized by the tuple $\langle N, \{\mathcal{M}_i\}, \mathcal{S}^{\text{obs}}, \mathcal{A}, P, R^{\text{soc}}, \gamma\rangle$. Here, N is the number of background agents, each agent i with a latent trait $m^i \in \mathcal{M}_i$. $\mathcal{S}^{\text{obs}}$ represents the *principal*'s comprehensive observation space, encompassing background agents' states $(s^1, ..s^N)$ that are visible to *principal* (like agents' locations) and global environment state s^E (like the distribution of resources). The *principal* acts by selecting a mechanism policy $\pi \sim \Pi(\pi|s^{\text{obs}})$, while each background agent i takes action following its policy $\phi_i(a^i|s^{\text{obs}}, \pi, m^i)$. The system transition function P describes how observations evolve given current observations and the deployed mechanism policy. That is

$$s_{t+1}^{\text{obs}} \sim P(s_{t+1}^{\text{obs}}|s_t^{\text{obs}}, \pi_t). \tag{1}$$

The *principal* infers the system transition function P and learns a mechanism policy π to maximize the expected cumulative social welfare

$$\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t^{soc}\right], \tag{2}$$

where $r^{\text{soc}} = \sum_i r^i$ is *principal*'s reward, named the social welfare, $r^i = R^i(s^{\text{obs}}, a^1, ..., a^N)$ is the reward of background agent i, and γ is the discount factor.

Thus, our formulation of AI-driven mechanism design presents two fundamental departures from classical mechanism design theory. 1) *Principal* has no prior knowledge of background agents' behavior patterns, which adaptively respond to the policies of *principal* and other background agents. *Principal* needs to conduct online inference of these patterns through interactions with them. 2) Agents exhibit heterogeneity, possessing diverse and persistent latent traits that individually shape their behaviors. Consequently, the *principal* must infer these distinct agent-specific traits and behavioral patterns from interaction trajectories, rather than relying on aggregated or homogeneous population models.

3.2 Social World Model-Augmented Mechanism Design Policy Learning

Our approach to AI-driven mechanism design problems within heterogeneous and dynamic multi-agent systems, termed Social World Model-Augmented Mechanism Design Policy Learning **(SWM-AP)**, leverages a model-based reinforcement learning framework. The core of our method consists of two primary, interconnected components: a **Social World Model (SWM)** that learns the complex dynamics of state transition, and a **Mechanism Design Policy** that learns to deploy optimal incentive mechanisms. This overall architecture is designed to address the challenge of unobservable agent traits and to maximize social welfare with a minimized sample.

Social World Model: SWM is tasked with learning a comprehensive model of the state transition function. Specifically, given the current observation s_t^{obs}, the deployed mechanism π_t, and an estimate of the agents' latent traits $\hat{\mathbf{m}} = (\hat{m}^1, ..., \hat{m}^N)$, SWM learns to predict the next observation s_{t+1}^{obs} and the immediate social welfare r_t^{soc}.

An important component of SWM is the Posterior Trait Tracker. Since the background agents' traits $\mathbf{m}$ are latent, accurate modeling of environment dynamics necessitates inferring these traits. The Posterior Trait Tracker is designed to infer these latent traits, $\hat{\mathbf{m}}_{\text{post}}$, by analyzing complete interaction trajectories $\tau = (s_0^{\text{obs}}, \pi_0, r_0^{\text{soc}}, \ldots, s_T^{\text{obs}}, \pi_T, r_T^{\text{soc}})$ collected during training. This module processes entire trajectory segments to capture long-term behavioral patterns indicative of the underlying traits. SWM, including its state prediction component, is then trained by minimizing the discrepancy

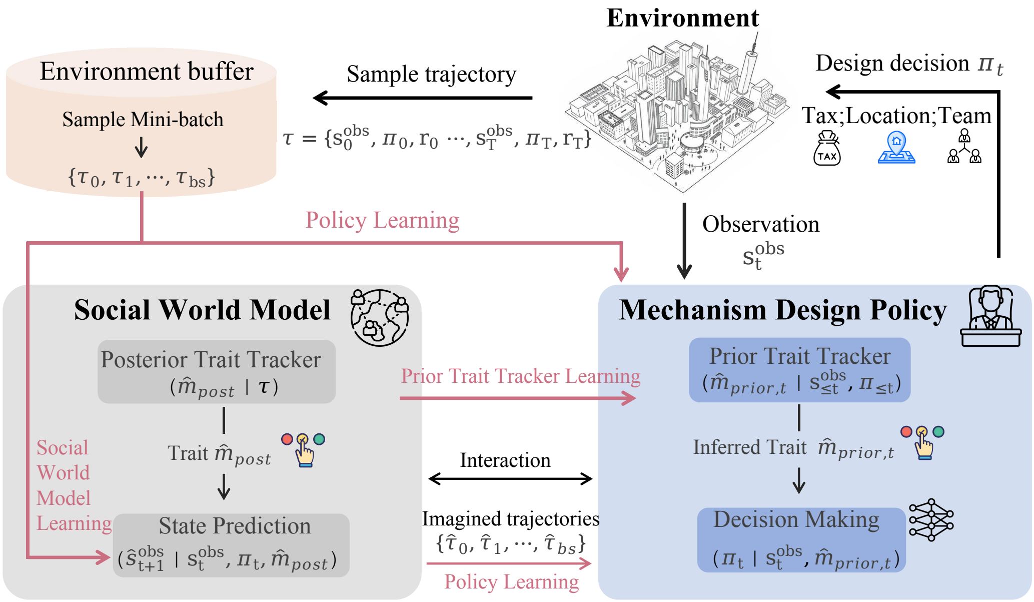

Environment
Environment buffer
Sample Mini-batch
$\{\tau_0, \tau_1, \cdots, \tau_{bs}\}$
Sample trajectory
$\tau = \{s_0^{obs}, \pi_0, r_0 \cdots, s_T^{obs}, \pi_T, r_T\}$
Design decision π_t
Tax;Location;Team
TAX
Observation
s_t^{obs}
Policy Learning
Social World Model
Posterior Trait Tracker
$(\hat{m}_{post} \mid \tau)$
Trait $\hat{m}_{post}$
Social World Model Learning
State Prediction
$(\hat{s}_{t+1}^{obs} \mid s_t^{obs}, \pi_t, \hat{m}_{post})$
Prior Trait Tracker Learning
Interaction
Imagined trajectories
$\{\hat{\tau}_0, \hat{\tau}_1, \cdots, \hat{\tau}_{bs}\}$
Policy Learning
Mechanism Design Policy
Prior Trait Tracker
$(\hat{m}_{prior,t} \mid s_{\leq t}^{obs}, \pi_{\leq t})$
Inferred Trait $\hat{m}_{prior,t}$
Decision Making
$(\pi_t \mid s_t^{obs}, \hat{m}_{prior,t})$

Figure 2: Algorithm diagram of SWM-AP. The **Social World Model (SWM)** utilizes a Posterior Trait Tracker to infer latent agent traits from full trajectories, which subsequently aid in predicting the future states of background agents. Concurrently, the **Mechanism Design Policy** employs a Prior Trait Tracker for real-time inference of agent traits based on partial history, informing its mechanism design. Interactions between the policy and the learned SWM, leveraging imagined trajectories, enhance sample efficiency for policy learning.

between its predicted future states and the actual observed states, utilizing these inferred traits $\hat{\mathbf{m}}_{\text{post}}$. The objective function of SWM can be expressed as:

$$J_{\text{SWM}}(\psi, \phi) = \mathbb{E}_{\tau \sim D}\left[\sum_t \|\hat{s}_{t+1}^{\text{obs}}(s_t^{\text{obs}}, \pi_t, \hat{\mathbf{m}}_{\text{post}}) - s_{t+1}^{\text{obs}}\|_2^2 + cD_{KL}\left(p(\hat{\mathbf{m}}_{\text{post}})||U\left(\mathbf{m}\right)\right)\right]. \quad (3)$$

Here, $J_{\text{SWM}}(\psi, \phi)$ represents the loss function for SWM with parameters ψ and the Posterior Trait Tracker with parameters ϕ. The expectation $\mathbb{E}_{\tau \sim D}$ is taken over trajectories τ sampled from a dataset D of past experiences. $\hat{s}_{t+1}^{\text{obs}}$ is the next state observation predicted by SWM, conditioned on the current state s_t^{obs}, mechanism π_t, and the traits $\hat{\mathbf{m}}_{\text{post}}$ inferred by the Posterior Trait Tracker. s_{t+1}^{obs} is the actual observed next state. The term $\| \cdot \|_2^2$ denotes the squared L2 norm (Euclidean distance), measuring the prediction error. $p(\hat{\mathbf{m}}_{\text{post}})$ is the probability output by the Posterior Trait Tracker, while $U(\mathbf{m})$ is the uniform distribution for every element in $\mathbf{m}$. c is the regularization coefficient. This joint optimization allows SWM to learn environment dynamics that are conditioned on a rich understanding of agent traits.

Mechanism Design Policy: The Mechanism Design Policy, denoted as $\Pi(\pi_t|s_t^{\text{obs}}, \hat{\mathbf{m}}_{\text{prior}})$, is responsible for selecting and deploying incentive mechanisms π_t to maximize the expected cumulative discounted social welfare, as defined in Equation 2. This policy is trained using PPO [29], interacting with the environment (either real or simulated by SWM) to gather experiences. The social welfare r_t^{soc} serves directly as the reward signal for policy updates.

A critical challenge during policy deployment is that complete future trajectories are unavailable for the Posterior Trait Tracker. To address this, the policy component incorporates a **Prior Trait Tracker**. This module is trained to perform real-time inference of background agents' traits, $\hat{\mathbf{m}}_{\text{prior}}$, based on the historically observed partial trajectory up to timestep t, i.e., $(s_0^{\text{obs}}, \pi_0, \ldots, s_t^{\text{obs}})$. The Prior Trait Tracker is trained in a supervised fashion, typically by minimizing the discrepancy between its predictions $\hat{\mathbf{m}}_{\text{prior},t}$ and the "ground truth" traits $\hat{\mathbf{m}}_{\text{post}}$ inferred by the Posterior Trait Tracker from complete trajectories during offline training. For example, a common objective is to minimize a cross-entropy loss if traits are categorical or a mean squared error if traits are continuous, at each step:

$$J_{\text{Prior}}(\xi) = \mathbb{E}_{\tau\sim D}\left[\sum_t L(\hat{\mathbf{m}}_{\text{prior},t}(s^{\text{obs}}_{\leq t}, \pi_{<t}, \mathbf{a}_{<t}), \hat{\mathbf{m}}_{\text{post}})\right]. \tag{4}$$

Here, $J_{\text{Prior}}(\xi)$ is the loss for the Prior Trait Tracker with parameters ξ. L is loss function comparing the prior tracker's estimate at time t, $\hat{\mathbf{m}}_{\text{prior},t}$, which is based on observations up to t, with the more accurate posterior estimate $\hat{\mathbf{m}}_{\text{post}}$ derived from the full trajectory.

During online interaction (policy execution), the inferred trait $\hat{\mathbf{m}}_{\text{prior}}$ from the Prior Trait Tracker is fed as input to both the Mechanism Design Policy Π (to inform its decision-making) and to SWM (when SWM is used for generating imagined trajectories). This allows the policy to adapt its mechanism design strategy dynamically based on its evolving understanding of the agents' latent traits. Furthermore, the policy interacts with the learned SWM to enhance sample efficiency. For instance, SWM can generate simulated rollouts under different candidate mechanisms, allowing the policy to be refined with significantly more data than direct environment interaction alone would permit. The Prior Trait Tracker's output can also be a probability distribution over possible traits, reflecting its prediction certainty, which the policy can leverage for more robust strategy learning [40]. This two-pronged approach, combining a world model that understands agent traits with a policy that leverages this understanding for real-time adaptation and efficient learning, forms the backbone of our method.

3.3 Theoretical Analysis

We derive the Evidence Lower Bound (ELBO) to provide a theoretical basis for unsupervised learning of latent agent traits from interaction data, within the episodic framework defined in subsection 3.1.

We denote the *principal*'s trajectory as $\tau = (s^{\text{obs}}_0, \pi_0, s^{\text{obs}}_1, \pi_1, \cdots, s^{\text{obs}}_{T-1}, \pi_{T-1}, s^{\text{obs}}_T)$, the joint actions as $\mathbf{a}_t = (a^1_t, \cdots, a^N_t)$, and the joint traits as $\mathbf{m} = (m^1, \cdots, m^N)$. We assume the joint distribution of τ and $\mathbf{m}$ follows a generative process:

$$p(\tau, \mathbf{m}) = p(\mathbf{m})p(s^{\text{obs}}_0)\prod_{t=0}^{T-1} p(\pi_t|s^{\text{obs}}_{\leq t})\int_{\mathbf{a}_t}\left(p(s^{\text{obs}}_{t+1}|s^{\text{obs}}_t, \pi_t, \mathbf{a}_t, \mathbf{m})\left(\prod_{i=1}^{N}\beta_i(a^i_t|s^i_t, m^i, \pi_t)\right)\right)d\mathbf{a}_t, \tag{5}$$

where $p(\mathbf{m})$ is the prior of the joint traits, and $\beta_i(\cdot|s^i_t, m^i, \pi_t)$ is the agent's fixed policy conditioned on its trait m^i the deployed mechanism π_t.

The world model $p_\psi(s^{\text{obs}}_{t+1}|s_t, \pi_t, \mathbf{m})$ approximates the dynamics $\int_{\mathbf{a}_t}\left(p(s^{\text{obs}}_{t+1}|s^{\text{obs}}_t, \pi_t, \mathbf{a}_t, \mathbf{m})\left(\prod_{i=1}^{N}\beta_i(a^i_t|s^i_t, \mathbf{m}, \pi_t)\right)\right)d\mathbf{a}_t$. The policy of the *principal* $\Pi_\theta(\pi_t|s_{\leq t})$ provides $p(\pi_t|s_{\leq t})$. Thus, the likelihood is estimated as

$$p_{\psi,\theta}(\tau|\mathbf{m}) = p(s^{\text{obs}}_0)\prod_{t=0}^{T-1}\Pi_\theta(\pi_t|s^{\text{obs}}_{\leq t})p_\psi(s^{\text{obs}}_{t+1}|s^{\text{obs}}_t, \pi_t, \mathbf{m}).$$

To generate new trajectories, we need to sample $\mathbf{m}$ from the posterior $p(\mathbf{m}|\tau)$ rather than the prior $p(\mathbf{m})$. However, $p(\mathbf{m}|\tau)$ is intractable, and we use the Posterior Trait Tracker $q_\phi(\mathbf{m}|\tau)$ to approximate it.

To maximize the log evidence $\log p(\tau)$, we need to maximize the ELBO:

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(\phi, \psi, \theta; \tau) = &\underbrace{\sum_{t=0}^{T-1}\mathbb{E}_{q_\phi(\mathbf{m}|\tau)}[\log p_\psi(s^{\text{obs}}_{t+1}|s^{\text{obs}}_t, \pi_t, \mathbf{m})]}_{\text{state prediction likelihood}} \\ &+ \underbrace{\sum_{t=0}^{T-1}\mathbb{E}_{q_\phi(\mathbf{m}|\tau)}[\log \Pi_\theta(\pi_t|s_{\leq t})]}_{\textit{principal}\text{ policy likelihood}} - \underbrace{D_{KL}(q_\phi(\mathbf{m}|\tau)||p(\mathbf{m}))}_{\text{regularization}}.\end{aligned} \tag{6}$$

As we use the same *principal* policy Π_θ to collect real trajectories and generate simulated trajectories, the *principal* policy likelihood is always maximized for ELBO. We minimize Equation 3 to maximize

Equation 6 under assumptions that $p(s_{t+1}^{\text{obs}}|s_t^{\text{obs}}, \pi_t, \mathbf{m})$ is Gaussian and each element of $p(\mathbf{m})$ is uniform. Please check the derivation details in Appendix A.

Algorithm 1 SWM-AP Learning framework

```
**Initialize:** Mechanism Design Policy $\Pi_\theta(\pi_t|s_t^{\text{obs}}, \hat{\mathbf{m}}_{\text{prior},t})$, Dynamic Model $M_\phi(\hat{s}_{t+1}^{\text{obs}}|s_t^{\text{obs}}, \pi_t, \hat{\mathbf{m}}_{\text{post}})$, Posterior Trait Tracker $q_\varphi(\hat{\mathbf{m}}_{\text{post}}|\tau)$, Prior Trait Tracker $p_\xi(\hat{\mathbf{m}}_{\text{prior},t}|H_t)$ where $H_t = (s_{\leq t}^{\text{obs}}, \pi_{<t}, \mathbf{a}_{<t})$, Environment, Model Datasets $D_{env}, D_{model}$
**for** $NEpochs$ **do**
    Collect real trajectories $\tau = (s_0^{\text{obs}}, \pi_0, r_0^{\text{soc}}, \dots, s_T^{\text{obs}}, \pi_T, r_T^{\text{soc}})$ in Environment using policy $\Pi_\theta$ and Prior Trait Tracker $p_\xi$. Store in $D_{env}$.
    Jointly train Posterior Trait Tracker $q_\varphi$ and Dynamic Model $M_\phi$ on dataset $D_{env}$, using objective based on Equation 3, implicitly training $q_\varphi$ to produce $\hat{\mathbf{m}}_{\text{post}}$.
    Train Prior Trait Tracker $p_\xi$ on dataset $D_{env}$ , using objective based on Equation 4 to align $p_\xi(\cdot|H_t)$ with $q_\varphi(\cdot|\tau)$.
    Generate imagined trajectories $\hat{\tau} = (\hat{s}_0^{\text{obs}}, \pi_0, \hat{r}_0^{\text{soc}}, \dots)$ using Dynamic Model $M_\phi$, policy $\Pi_\theta$, and Posterior Trait Tracker $p_\xi$. Store in $D_{model}$.
    Optimize policy $\Pi_\theta$ using data from $D_{env}$ and $D_{model}$ , maximizing objective Equation 2 using PPO on combined data.
**end for**
**Return:** Policy $\Pi_\theta$, SWM $p_\psi$
```

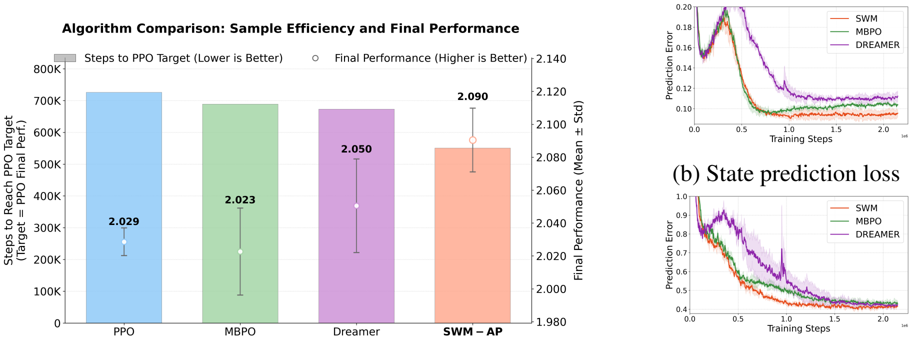
4 Experiments

4.1 Facility Location

We designed a facility location game to examine the effectiveness of the methodology, where we developed learning strategies to enhance the capability of higher-level agents in selecting optimal facility construction locations for the background populations of rule-based agents.

Environment Setting: In the facility location game, the mechanism designer is tasked with selecting appropriate facility locations for multiple agents distributed across a map. Different agents exhibit heterogeneous preferences regarding facility locations. Our approach achieves optimal mechanism design by learning dynamic mechanism design strategies to maximize the collective reward of multiple low-level agents. The environment is configured as a matrix where each agent maintains a fixed global position at the beginning of each round, representing their permanent residence in real-world scenarios. The mechanism designer determines facility locations each round with a fixed total number of facilities, where each location configuration influences the total visitation frequency of low-level agents to facilities. The reward is defined as the summation of visitation frequencies from low-level agents to facilities. This experimental setup corresponds to the classical facility location game in mechanism design theory. Specifically, we implement a configuration with five facilities and eight agents distributed across an 8×8 grid.

Performance Analysis: We evaluate our proposed **SWM-AP** method against several baselines: the model-based reinforcement learning algorithms Dreamer and MBPO, and the model-free RL algorithm PPO. The comparative results are presented in Figure 3. Figure 3a utilizes a dual y-axis plot to illustrate two key performance aspects: sample efficiency and final converged reward. On this plot, the circles, aligned with the right y-axis, represent the mean final reward (± standard deviation) achieved by each algorithm upon convergence. The bars, corresponding to the left y-axis, indicate the number of training steps required for each method to reach a predefined performance target, specifically PPO's final converged reward. The results demonstrate that model-based methods generally exhibit superior sample efficiency compared to their model-free counterparts. Notably, our method not only achieves the highest sample efficiency, requiring the fewest training steps to meet the performance target, but also attains the highest final converged reward among all evaluated algorithms. Figure 3b and Figure 3c display the learning loss curves for system states and immediate rewards, respectively. We find that our SWM achieves more accurate predictions for both metrics when compared to other baselines.



(a) Facility location: Sample efficiency and final performance

(b) State prediction loss

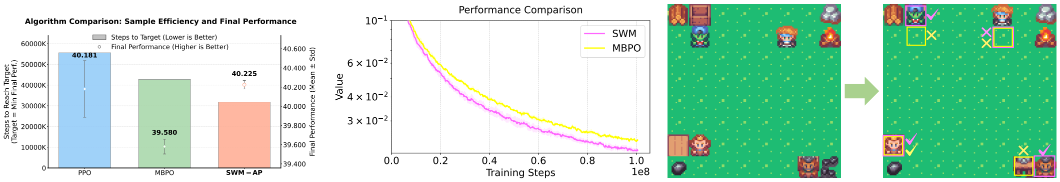
(c) Reward prediction loss

Figure 3: Facility location performance analysis. (a) Comparison of sample efficiency and final converged performance. (b) State prediction loss and (c) Reward prediction loss curves for our SWM compared to baselines.

4.2 Team Structure Optimization

Team structure optimization, where the team structure of background agents can be dynamically adjusted by *principal*, is another widely studied mechanism design problem. We conduct the experiments in AdaSociety [13], a highly customizable multi-agent environment supporting dynamic social relationships and heterogeneous agents with open-ended tasks. By controlling the relationships between background agents, *principal* aims to maximize the collective reward of background agents.

Environment Setting: The environment consists of an 8×8 grid with four types of basic resources (10 units each, positioned at the corners), where two basic resources can be converted into one advanced resource (valued at 5, compared to 1 for basic resources). Four agent types exist, each capable of producing a specific basic resource but unable to store it. Instead, they can store one other predefined resource type. Agents form teams of arbitrary size, with each agent restricted to one team, and incur a maintenance cost of $0.05(x-1)$ per step, where x is the team size. Agents produce resources matching their type and only store resources to earn rewards if a teammate can produce their storable types. In each episode, four background agents are initialized with randomly assigned types. *Principal* observes the current map without knowing agents' types, and then reassigns the team structure every 10 steps starting at step 5, aiming to maximize total group reward. Each episode lasts for 50 timesteps. The task challenges *principal* to optimize collective efficiency in a resource-constrained multi-agent system, where *principal* must balance production, storage dependencies, and team coordination costs.



(a) Performance

(b) State loss

(c) An example

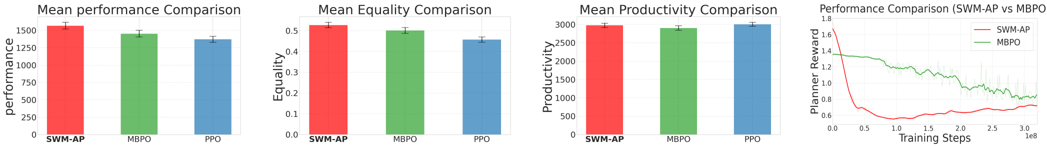
Figure 4: Performance comparison in team structure optimization

Performance Analysis: We evaluate our **SWM-AP** method against representative model-based and model-free RL baselines in the dynamic team structure optimization task within AdaSociety. The comparative results are presented in Figure 4. Figure 4a highlights key performance metrics: sample efficiency (bars, left y-axis), indicating training steps to reach a target performance, and final converged group reward (circles/points, right y-axis, with ± standard deviation). Consistent with findings in other domains, model-based approaches demonstrate superior sample efficiency. Notably, our method not only achieves the highest sample efficiency, requiring the fewest steps to reach the target, but also secures the highest final converged group reward, showcasing its effectiveness in optimizing team configurations under resource constraints and dynamic reassignments. Figure 4b depicts the learning curves for the predictive components of the world model. These plots indicate that our SWM, which explicitly infers and uses agent traits to model team dynamics, achieves lower prediction errors compared to model-based baselines. Figure 4c illustrates an example of SWM and MBPO in this environment. Both algorithms take the current state (left sub-graph) as input and predict the location of each agent in the next timestep. These predictions are shown in the right sub-graph, with our SWM's prediction marked in pink and MBPO's in yellow. The right sub-graph also displays the actual location of the agent in the next timestep for comparison. Overall, SWM's predictions are more accurate. For instance, for the agent in the lower right corner, SWM correctly predicts that the agent will move to the grid on the right, where resources are located, while MBPO predicts that the agent will stay in its current position, where the manufacturing of advanced resources is taking place. This accuracy is likely because SWM has a better understanding of background agents' traits, enabling it to more precisely infer these agents' action plans.

4.3 Tax Adjustment

In this experimental setup, low-level agents are trained using reinforcement learning. As a result, the planner interacts with a continually adapting and improving population of agents, making the environment increasingly complex over time.

Environment Setting: In AI-Economist [39], background agents engage in activities such as collecting materials (specifically wood and stone) to construct houses in exchange for income. They can also trade resources on the market. Agents possess varying levels of skills in house construction, and their primary objective is to maximize individual utility. This utility is positively influenced by income but negatively affected by labor effort. Therefore, agents make decisions by considering several economic variables, including their construction skills, resource endowments, and applicable tax rates. These factors influence both their work and consumption choices. *Principal*, whose role is to design tax policies, seeks to enhance social welfare by balancing overall economic productivity with income equality. Notably, *principal* lacks visibility into the agents' specific construction skills. For this experiment, the environment consists of four low-level agents operating within a 25×25 map. Each episode spans 1000 time steps, while *principal* updates tax policies every 100 steps.



(a) Overall performance (b) Equality (c) Productivity (d) State loss

Figure 5: Performance comparison in AI-Economist

Performance Analysis: We assess the effectiveness of our SWM-AP by comparing it with two baseline algorithms: the model-based MBPO and the model-free PPO. The evaluation is based on three social metrics recorded after 1,000 agent steps: *equality*, *productivity*, and their product, *equality* × *productivity*. The product metric serves as an indicator of overall social welfare within the simulated environment. The comparative results are presented in Figure 5. As shown, our method exhibits effective control over taxation, enabling *principal* to optimize social welfare. Notably, our method attains a level of productivity comparable to that of PPO while significantly enhancing equality. This suggests that our SWM is capable of promoting a more equitable society without compromising economic output. In contrast, MBPO yields suboptimal performance. Although it attempts to increase equality, this comes at the cost of a marked decline in productivity, ultimately resulting in lower overall social welfare relative to our method. We attribute the superior performance of our method to its ability to reduce state prediction error more efficiently during the early phases of training (as shown in Figure 5d). This improved predictive accuracy facilitates more effective *principal* optimization, thereby leading to better overall outcomes.

5 Conclusion

This paper proposes a social world model-augmented mechanism design policy learning method, named SWM-AP, which employs unsupervised learning to infer the hidden trait of background agents, thereby enhancing the prediction of group dynamics and mechanism design policy learning. Experimental results in facility location, team structure optimization, and taxation setting tasks demonstrate that our method outperforms both model-based and model-free reinforcement learning approaches. Our approach inspires new directions for world model research towards more complex and realistic social world.

Limitations and Future Work: Current limitations include the scalability of SWM to extremely large-scale systems and the challenge of ensuring the direct interpretability of all inferred latent traits. Future work will focus on developing more scalable SWM architectures, potentially leveraging structured priors, and on enhancing trait interpretability through techniques like disentangled representation learning.

Acknowledgments and Disclosure of Funding

This work is supported by the National Science and Technology Major Project (No. 2022ZD0114904).

References

[1] Mark Armstrong. Interactions between competition and consumer policy. *Competition Policy International*, 4(1), 2008.

[2] Ruiqing Chen, Xiaoyuan Zhang, Yali Du, Yifan Zhong, Zheng Tian, Fanglei Sun, and Yaodong Yang. Off-agent trust region policy optimization. In *International Joint Conference on Artificial Intelligence*, 2024.

[3] Zhixun Chen, Zijing Shi, Yaodong Yang, Meng Fang, and Yali Du. Hierarchical multi-agent framework for dynamic macroeconomic modeling using large language models. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pages 2460–2462, 2025.

[4] Mingyu Guo, Diksha Goel, Guanhua Wang, Runqi Guo, Yuko Sakurai, and Muhammad Ali Babar. Mechanism design for public projects via three machine learning based approaches. *Autonomous Agents and Multi-Agent Systems*, 38(1):16, 2024.

[5] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[6] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

[7] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

[8] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[9] Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023.

[10] Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022.

[11] Zehong Hu, Yitao Liang, Jie Zhang, Zhao Li, and Yang Liu. Inference aided reinforcement learning for incentive mechanism design in crowdsourcing. *Advances in Neural Information Processing Systems*, 31, 2018.

[12] Yizhe Huang, Anji Liu, Fanqi Kong, Yaodong Yang, Song-Chun Zhu, and Xue Feng. Efficient adaptation in mixed-motive environments via hierarchical opponent modeling and planning. In *International Conference on Machine Learning*, pages 20004–20022. PMLR, 2024.

[13] Yizhe Huang, Xingbo Wang, Hao Liu, Fanqi Kong, Aoyang Qin, Min Tang, Xiaoxi Wang, Song-Chun Zhu, Mingjie Bi, Siyuan Qi, et al. Adasociety: An adaptive environment with social structures for multi-agent decision-making. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[14] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

[15] Fanqi Kong, Xiaoyuan Zhang, Xinyu Chen, Yaodong Yang, Song-Chun Zhu, and Xue Feng. Enhancing llm-based social bot via an adversarial learning framework. *arXiv preprint arXiv:2508.17711*, 2025.

[16] Raphael Koster, Jan Balaguer, Andrea Tacchetti, Ari Weinstein, Tina Zhu, Oliver Hauser, Duncan Williams, Lucy Campbell-Gillingham, Phoebe Thacker, Matthew Botvinick, et al. Human-centred mechanism design with democratic ai. *Nature Human Behaviour*, 6(10):1398–1407, 2022.

[17] Raphael Koster, Miruna Pîslar, Andrea Tacchetti, Jan Balaguer, Leqi Liu, Romuald Elie, O Hauser, Karl Tuyls, Matt Botvinick, and Christopher Summerfield. Using deep reinforcement-learning to discover a dynamic resource allocation policy that promotes sustainable human exchange. *Nature Communications*, 2025.

[18] Boxiang Lyu, Zhaoran Wang, Mladen Kolar, and Zhuoran Yang. Pessimism meets vcg: Learning dynamic mechanism design via offline reinforcement learning. In *International Conference on Machine Learning*, pages 14601–14638. PMLR, 2022.

[19] Chengdong Ma, Aming Li, Yali Du, Hao Dong, and Yaodong Yang. Efficient and scalable reinforcement learning for large-scale network control. *Nature Machine Intelligence*, 6(9):1006–1020, 2024.

[20] Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

[21] James A Mirrlees. The theory of optimal taxation. *Handbook of mathematical economics*, 3:1197–1249, 1986.

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

[23] Roger B Myerson. Optimal coordination mechanisms in generalized principal–agent problems. *Journal of mathematical economics*, 10(1):67–81, 1982.

[24] Hyungho Na and Il-chul Moon. Lagma: Latent goal-guided multi-agent reinforcement learning. *arXiv preprint arXiv:2405.19998*, 2024.

[25] Noam Nisan and Amir Ronen. Algorithmic mechanism design. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 129–140, 1999.

[26] Julien Perolat, Joel Z Leibo, Vinicius Zambaldi, Charles Beattie, Karl Tuyls, and Thore Graepel. A multi-agent reinforcement learning model of common-pool resource appropriation. *Advances in neural information processing systems*, 30, 2017.

[27] Siyuan Qi, Shuo Chen, Yexin Li, Xiangyu Kong, Junqi Wang, Bangcheng Yang, Pring Wong, Yifan Zhong, Xiaoyuan Zhang, Zhaowei Zhang, et al. Civrealm: A learning and reasoning odyssey in civilization for decision-making agents. *arXiv preprint arXiv:2401.10568*, 2024.

[28] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Learning humanoid locomotion with transformers. *CoRR*, 2023.

[29] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[30] Weiran Shen, Binghui Peng, Hanpeng Liu, Michael Zhang, Ruohan Qian, Yan Hong, Zhi Guo, Zongyao Ding, Pengjun Lu, and Pingzhong Tang. Reinforcement mechanism design: With applications to dynamic pricing in sponsored search auctions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 2236–2243, 2020.

[31] Tianmin Shu and Yuandong Tian. M3rl: Mind-aware multi-agent management reinforcement learning. *arXiv preprint arXiv:1810.00147*, 2018.

[32] Pingzhong Tang. Reinforcement mechanism design. In *IJCAI*, pages 5146–5150, 2017.

[33] Aravind Venugopal, Stephanie Milani, Fei Fang, and Balaraman Ravindran. Mabl: Bi-level latent-variable world model for sample-efficient multi-agent reinforcement learning. *arXiv preprint arXiv:2304.06011*, 2023.

[34] Xihuai Wang, Zhicheng Zhang, and Weinan Zhang. Model-based multi-agent reinforcement learning: Recent progress and prospects. *arXiv preprint arXiv:2203.10603*, 2022.

[35] Annie Xie, Dylan Losey, Ryan Tolsma, Chelsea Finn, and Dorsa Sadigh. Learning latent representations to influence multi-agent interaction. In *Conference on robot learning*, pages 575–588. PMLR, 2021.

[36] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

[37] Hongxin Zhang, Zeyuan Wang, Qiushi Lyu, Zheyuan Zhang, Sunli Chen, Tianmin Shu, Behzad Dariush, Kwonjoon Lee, Yilun Du, and Chuang Gan. Combo: compositional world models for embodied multi-agent cooperation. *arXiv preprint arXiv:2404.10775*, 2024.

[38] Xiaoyuan Zhang, Xinyan Cai, Bo Liu, Weidong Huang, Song-Chun Zhu, Siyuan Qi, and Yaodong Yang. Differentiable information enhanced model-based reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22605–22613, 2025.

[39] Stephan Zheng, Alexander Trott, Sunil Srinivasa, Nikhil Naik, Melvin Gruesbeck, David C Parkes, and Richard Socher. The ai economist: Improving equality and productivity with ai-driven tax policies. *arXiv preprint arXiv:2004.13332*, 2020.

[40] Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. *arXiv preprint arXiv:1910.08348*, 2019.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions: a novel Social World Model -augmented mechanism design policy learning framework that addresses agent heterogeneity and improves sample efficiency. These claims are directly supported by the proposed methodology and are intended to be validated through experimental results in tasks like tax policy, team formation, and facility location. The scope is defined as AI-driven mechanism design in complex social systems.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated "Limitations and Future Work" part 5 within the Conclusion. This subsection explicitly discusses limitations such as the scalability of the Social World Model (SWM) to extremely large-scale systems and challenges related to the interpretability of inferred latent traits. It also outlines corresponding future research directions to address these points.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper presents a theoretical analysis, including the derivation of the Evidence Lower Bound (ELBO) for the unsupervised learning of latent agent traits 3.3.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper details the experimental setup for each task (facility location, team structure optimization, taxation) including environment configurations, agent details (if applicable), and evaluation metrics. Hyperparameter settings for our method and baselines, as well as architectural details of the models, are provided in appendix, supplemental materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We are currently organizing and refining our source code and experimental environments. We anticipate making them available at a later stage once the organization process is complete.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specifies the training procedures, including the number of epochs/timesteps, and batch sizes for our method and baselines. Details on hyperparameter selection and architectural choices for neural networks are provided in the experimental sections and further elaborated in the appendix, supplemental materials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Experimental results presented in figures 3, 4, 5 include error bars over multiple independent runs. This provides an indication of the variability and consistency of the results. The method for calculating these error bars is based on aggregating results from these multiple runs.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Information regarding the computational resources used for experiments, including the type of GPUs (e.g., NVIDIA RTX 3090) and CPUs, in the appendix, supplemental materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research presented in this paper involves the development and simulation-based evaluation of algorithms for mechanism design. It does not involve direct interaction with human subjects, collection of personally identifiable information, or applications with immediate high-risk societal implications in its current form. We have adhered to standard academic practices and believe our work conforms to the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses the potential positive societal impacts of improved mechanism design, such as more efficient resource allocation, optimized public policies (e.g., taxation), and enhanced team coordination, leading to better social welfare. We also briefly acknowledge potential negative societal impacts in the "Limitations and Future Work" section by touching upon the need for interpretability and robustness, as poorly understood or biased SWMs could lead to suboptimal or unfair mechanisms if deployed without caution. Further discussion on mitigating misuse.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The models developed in this research (Social World Model and Mechanism Design Policy) are trained from scratch for specific simulated mechanism design tasks and do not fall into the category of large pretrained models with immediate high-risk misuse potential like large language models or image generators. We do not use scraped datasets that might pose privacy or safety risks. While any powerful simulation tool could theoretically be misused, the current research focuses on foundational algorithmic development within controlled simulated environments.

    Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer:[Yes]

    Justification: Existing assets used are properly credited through citations to their original publications in the bibliography.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper introduces a novel methodological framework and algorithms, but does not introduce new standalone datasets or pre-trained models as primary contributions for release beyond the described research. The focus is on the algorithmic approach and its empirical validation.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This research does not involve crowdsourcing or direct experiments with human subjects. All experiments are conducted in simulated multi-agent environments.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This research does not involve human subjects, and therefore, IRB approval was not required or sought.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core methodology presented in this paper for Social World Model-Augmented Mechanism Design Policy Learning does not involve the use of Large Language Models (LLMs) as an important, original, or non-standard component. Any LLM usage was limited to aiding in writing and editing, without impacting the core scientific contributions or methods.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A ELBO Derivation

We denote the *principal*'s trajectory as $\tau = (s_0^{\text{obs}}, \pi_0, s_1^{\text{obs}}, \pi_1, \cdots, s_{T-1}^{\text{obs}}, \pi_{T-1}, s_T^{\text{obs}})$, the joint actions as $\mathbf{a}_t = (a_t^1, \cdots, a_t^N)$, and the joint traits as $\mathbf{m} = (m^1, \cdots, m^N)$. We assume the joint distribution of τ and $\mathbf{m}$ follows a generative process:

$$p(\tau, \mathbf{m}) = p(\mathbf{m})p(s_0^{\text{obs}}) \prod_{t=0}^{T-1} p(\pi_t|s_{\leq t}^{\text{obs}}) \int_{\mathbf{a}_t} \left(p(s_{t+1}^{\text{obs}}|s_t^{\text{obs}}, \pi_t, \mathbf{a}_t, \mathbf{m}) \left(\prod_{i=1}^{N} \beta_i(a_t^i|s_t^i, m^i, \pi_t) \right) \right) d\mathbf{a}_t, \tag{7}$$

where $p(\mathbf{m})$ is the prior of the joint traits, and $\beta_i(\cdot|s_t^i, m^i, \pi_t)$ is the agent's fixed policy conditioned on its trait m^i the deployed mechanism π_t.

The world model $p_\psi(s_{t+1}^{\text{obs}}|s_t, \pi_t, \mathbf{m})$ approximates the dynamics $\int_{\mathbf{a}_t} \left(p(s_{t+1}^{\text{obs}}|s_t^{\text{obs}}, \pi_t, \mathbf{a}_t, \mathbf{m}) \left(\prod_{i=1}^{N} \beta_i(a_t^i|s_t^i, \mathbf{m}, \pi_t) \right) \right) d\mathbf{a}_t$. The policy of the *principal* $\Pi_\theta(\pi_t|s_{\leq t})$ provides $p(\pi_t|s_{\leq t})$. Thus, the likelihood is estimated as

$$p_{\psi,\theta}(\tau|\mathbf{m}) = p(s_0^{\text{obs}}) \prod_{t=0}^{T-1} \Pi_\theta(\pi_t|s_{\leq t}^{\text{obs}}) p_\psi(s_{t+1}^{\text{obs}}|s_t^{\text{obs}}, \pi_t, \mathbf{m}).$$

To generate new trajectories, we need to sample $\mathbf{m}$ from the posterior $p(\mathbf{m}|\tau)$ rather than the prior $p(\mathbf{m})$. However, $p(\mathbf{m}|\tau)$ is intractable, and we use the Posterior Trait Tracker $q_\phi(\mathbf{m}|\tau)$ to approximate it.

Here, we derive a lower bound for the log evidence $\log p_{\psi,\phi,\theta}(\tau)$:

$$\begin{aligned}
\log p_{\psi,\phi,\theta}(\tau) &= \mathbb{E}_{q_\phi(\mathbf{m}|\tau)}[\log p_{\psi,\theta}(\tau)] \\
&= \mathbb{E}_{q_\phi(\mathbf{m}|\tau)}[\log \frac{p_{\psi,\theta}(\tau, \mathbf{m})}{p_{\psi,\theta}(\mathbf{m}|\tau)}] \\
&= \mathbb{E}_{q_\phi(\mathbf{m}|\tau)}[\log \frac{p_{\psi,\theta}(\tau, \mathbf{m})}{q_\phi(\mathbf{m}|\tau)}] + \mathbb{E}_{q_\phi(\mathbf{m}|\tau)}[\log \frac{q_\phi(\mathbf{m}|\tau)}{p_{\psi,\theta}(\mathbf{m}|\tau)}] \\
&= \mathbb{E}_{q_\phi(\mathbf{m}|\tau)}[\log \frac{p_{\psi,\theta}(\tau, \mathbf{m})}{q_\phi(\mathbf{m}|\tau)}] + D_{KL}(q_\phi(\mathbf{m}|\tau)||p_{\psi,\theta}(\mathbf{m}|\tau)) \\
&\geq \mathbb{E}_{q_\phi(\mathbf{m}|\tau)}[\log \frac{p_{\psi,\theta}(\tau, \mathbf{m})}{q_\phi(\mathbf{m}|\tau)}] \\
&= \mathbb{E}_{q_\phi(\mathbf{m}|\tau)}[\log p_{\psi,\theta}(\tau, \mathbf{m}) - \log q_\phi(\mathbf{m}|\tau)] \\
&= \mathbb{E}_{q_\phi(\mathbf{m}|\tau)}[\log p_{\psi,\theta}(\tau|\mathbf{m}) + \log p(\mathbf{m}) - \log q_\phi(\mathbf{m}|\tau)] \\
&= \mathbb{E}_{q_\phi(\mathbf{m}|\tau)}[\log p_{\psi,\theta}(\tau|\mathbf{m})] - \mathbb{E}_{q_\phi(\mathbf{m}|\tau)}[\log \frac{q_\phi(\mathbf{m}|\tau)}{p(\mathbf{m})}] \\
&= \mathbb{E}_{q_\phi(\mathbf{m}|\tau)}[\log p_{\psi,\theta}(\tau|\mathbf{m})] - D_{KL}(q_\phi(\mathbf{m}|\tau)||p(\mathbf{m})) \\
&= \log p(s_0^{\text{obs}}) + \underbrace{\sum_{t=0}^{T-1} \mathbb{E}_{q_\phi(\mathbf{m}|\tau)}[\log p_\psi(s_{t+1}^{\text{obs}}|s_t^{\text{obs}}, \pi_t, \mathbf{m})]}_{\text{State Prediction Likelihood}} \\
&\quad + \underbrace{\sum_{t=0}^{T-1} \mathbb{E}_{q_\phi(\mathbf{m}|\tau)}[\log \beta_\theta(\pi_t|s_{\leq t}^{\text{obs}})]}_{\textit{principal} \text{ policy likelihood}} - \underbrace{D_{KL}(q_\phi(\mathbf{m}|\tau)||p(\mathbf{m}))}_{\text{Regularization Term}}.
\end{aligned}$$

B Trait Inference Confusion Matrices

In our framework, a "trait" (m) represents a persistent, intrinsic characteristic of an agent. In tasks such as AdaSociety, it manifests as an agent's inherent production capability, while the SWM-AP method infers these unobservable attributes (e.g., skills, preferences, or risk attitudes) from agent interaction trajectories, shaping their behavior. This trait is not directly encapsulated in the observation space (s_t), rendering it inaccessible to the principal. We argue that the necessity of modeling such traits is directly tied to the degree of observational ambiguity in a system. In real-world scenarios, a principal (e.g., a government or platform) must make decisions under conditions of incomplete and ambiguous information. An individual's observable state (s_t) at any given moment, such as their current location or recent purchase, is often a highly ambiguous signal of their underlying trait (m), such as risk aversion, long-term preferences, or intrinsic skills. For instance, two individuals might be observed in the same location (s_t), but one is a risk-averse local resident (trait m_1) while the other is an adventurous tourist (trait m_2). Their immediate responses (a_t) and future state (s_{t+1}) to a new incentive (e.g., dynamic pricing) will diverge drastically, and predicting this divergence is impossible without inferring their underlying traits.

To test this hypothesis and explore trait interpretability under observational ambiguity, we conducted diagnostic experiments in the AdaSociety task. In standard training, the Posterior Trait Tracker produces a confusion matrix with clear diagonal dominance, indicating successful differentiation between agent types(Figure 6, left). However, non-trivial off-diagonal values suggest room for improvement, attributable to "modeling shortcuts". In later episode stages, the environment's predictability allows the model to rely on state history rather than precise trait inference, achieving low prediction error. To address this, we trained the tracker exclusively on high-ambiguity initial states, where agents' fixed starting positions provide minimal clues about their randomly assigned types. The resulting confusion matrix (Figure 6, right) exhibits stronger diagonal dominance. This demonstrates that SWM-AP's ability to learn interpretable traits is significantly enhanced under high-ambiguity conditions, which directly addresses persistent informational asymmetry in real-world scenarios.

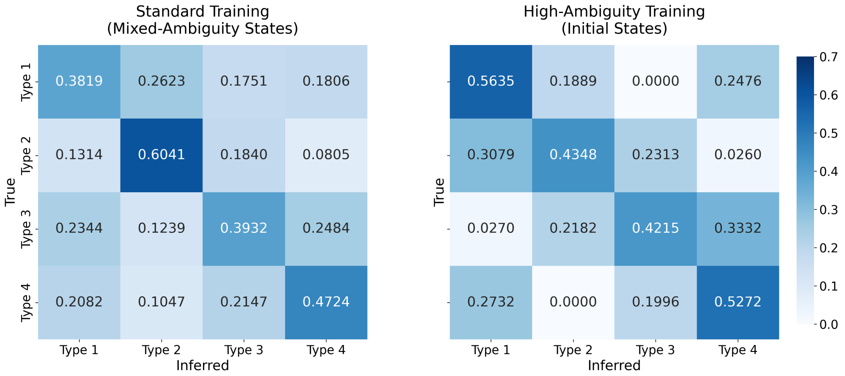


Figure 6: Confusion matrices for trait inference in AdaSociety. Left: Standard training on mixed-ambiguity states, showing moderate diagonal dominance due to modeling shortcuts in low-ambiguity late-episode states. Right: High-ambiguity training on initial states, demonstrating improved trait disentanglement with stronger diagonal dominance.

C Performance and Efficiency Results

To assess the scalability of SWM-AP in larger multi-agent settings, we extended the Facility Location task to 32 agents on a 7x7 grid with 5 placeable facilities. Each agent was randomly assigned one of two unobservable latent traits at the start of each episode, governing their behavior based on distance and congestion, with fixed home locations. The principal made 5 mechanism design decisions per episode. Results, averaged over 3 independent runs with different random seeds, are presented in Table 1. Efficiency is measured as the number of training steps required to reach MBPO's final performance (reward of 6.43), with lower values indicating better sample efficiency.

Table 1: Performance and efficiency results for the 32-agent Facility Location task.

Algorithm	Final Reward (Mean ± Std)	Steps to MBPO Final Perf.
PPO	6.55 ± 0.03	353,600
MBPO	6.43 ± 0.04	433,600
Dreamer	6.57 ± 0.06	300,800
SWM-AP	6.62 ± 0.06	274,667

These results demonstrate that SWM-AP achieves a superior sample efficiency compared to baselines.

D Computational Benchmarks

We report the runtime and memory usage benchmarked against the number of agents. The results indicate that both the runtime and memory footprint of our method fall within an acceptable range.

Table 2: Computational benchmarks for the Facility Location task across different agent numbers, per 100k training steps.

Agents	Training Time (hrs)	Memory Footprint (MB)
2	0.91	538
4	1.15	618
8	1.93	898
16	3.45	1786
32	5.25	5104

E Experimental Details

This appendix provides essential details for reproducibility. Key configurations and hyperparameters are summarized below. Table 3 summarizes crucial parameters for the experimental environments and core algorithms.

Table 3: Key Environment and Algorithm Configurations.

Category	Parameter	Facility Location	Team Structure Optimization	Tax Adjustment
Environment Specifics				
	Env. Source	Matrix	AdaSociety	AI Economist
	Agent Count	8	4	4
	Latent Trait Count.	256	4	4
	Mechanism Action	Select a point from Map(8*8)	Assign a team structures among 14 different types	Set a tax rate for each of the 7 tax brackets
	Episode Length	5	50	1000 (for agents), 10 (for planner)
SWM-AP: Social World Model (SWM)				
	Latent Inference Arch.	MLP (L:2, H:512)	MLP (L:2, H:512) + LSTM (L:1, H:512) + MLP (L:2, H:512)	MLP (L:2, H:512) + LSTM (L:1, H:512) + MLP (L:2, H:512)
	Dynamics Predict. Arch.	MLP (L:3, H:256)	GCN (L:3, H:[64, 128]) + MLP (L:2, H:128)	GCN (L:3, H:[64, 128]) + MLP (L:2, H:128)
	SWM Optimizer & LR	Adam, 10^{-3}	Adam, 10^{-3}	Adam, 10^{-3}
SWM-AP: Mechanism Design Policy (PPO based)				
	Policy/Value Arch.	MLP (L:2, H:128)	GCN (L:3, H:[64, 64]) + MLP (L:2, H:256)	MLP (L:2, H:256) + LSTM(L:1,H:256) + MLP (L:1,H:256)
	Optimizer & LR	Adam, 2.5×10^{-4}	Adam, 5×10^{-4}	Adam, 1×10^{-4}
	Discount (γ)	0.99	0.99	0.99
	Imagined Rollout (SWM)	5 steps	50 steps	1000 steps
Baselines: PPO				
	Policy/Value Arch.	MLP (L:2, H:128)	GCN (L:3, H:[64, 64]) + MLP (L:2, H:256)	MLP (L:2, H:256) + LSTM(L:1,H:256) + MLP (L:1,H:256)
	Optimizer & LR	Adam, 2.5×10^{-4}	Adam, 5×10^{-4}	Adam, 1×10^{-4}
	Discount (γ)	0.99	0.99	0.99
Baselines: MBPO				
	Policy/Value Arch.	MLP (L:2, H:128)	GCN (L:3, H:[64, 64]) + MLP (L:2, H:256)	GCN(L:3, H:[64, 64]) + MLP (L:2, H:256)
	Optimizer & LR	Adam, 2.5×10^{-4}	Adam, 5×10^{-4}	Adam, 5×10^{-4}
	Discount (γ)	0.99	0.99	0.99
General Training & Compute				
	Total Timesteps	10^{6}	1×10^{8}	5×10^{8}
	Num. Random Seeds	3	3	3
	Error Bars	± SEM over 3 runs	± SEM over 3 runs	± SEM over 3 runs
	GPUs Used	NVIDIA RTX 3090 (1 per run)	NVIDIA RTX 3090 (1 per run)	NVIDIA A100 (1 per run)

Further Details on SWM-AP: SWM is trained to minimize Mean Squared Error for dynamics prediction, potentially with an additional VAE-like loss for trait inference if applicable. **Further Details on Baselines:** Model-free baselines like PPO implemented with standard configurations. For PPO, a clipping epsilon of 0.2 was used.