
Grounded Answers for Multi-agent Decision-making Problem through Generative World Model

Zeyang Liu

zeyang.liu@stu.xjtu.edu.cn

Xinrui Yang

xinrui.yang@stu.xjtu.edu.cn

Shiguang Sun

ssg2019@stu.xjtu.edu.cn

Long Qian

qianlongym@stu.xjtu.edu.cn

Lipeng Wan

wanlipeng77@xjtu.edu.cn

Xingyu Chen*

chenxingyu_1990@xjtu.edu.cn

Xuguang Lan*

xglan@mail.xjtu.edu.cn

**National Key Laboratory of Human-Machine Hybrid Augmented Intelligence
National Engineering Research Center for Visual Information and Application
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China**

Abstract

Recent progress in generative models has stimulated significant innovations in many fields, such as image generation and chatbots. Despite their success, these models often produce sketchy and misleading solutions for complex multi-agent decision-making problems because they miss the trial-and-error experience and reasoning as humans. To address this limitation, we explore a paradigm that integrates a language-guided simulator into the multi-agent reinforcement learning pipeline to enhance the generated answer. The simulator is a world model that separately learns dynamics and reward, where the dynamics model comprises an image tokenizer as well as a causal transformer to generate interaction transitions autoregressively, and the reward model is a bidirectional transformer learned by maximizing the likelihood of trajectories in the expert demonstrations under language guidance. Given an image of the current state and the task description, we use the world model to train the joint policy and produce the image sequence as the answer by running the converged policy on the dynamics model. The empirical results demonstrate that this framework can improve the answers for multi-agent decision-making problems by showing superior performance on the training and unseen tasks of the StarCraft Multi-Agent Challenge benchmark. In particular, it can generate consistent interaction sequences and explainable reward functions at interaction states, opening the path for training generative models of the future.

1 Introduction

Recent progress in generative artificial intelligence with models capable of generating creative content has shown attractive prospects for real-world applications, such as image generation (Takagi & Nishimoto, 2023), embodied agents (Brohan et al., 2023b), and chatbots (Köpf et al., 2024). Most generative models attempt to directly obtain the answer by training on natural language or image datasets and inserting decomposed reasoning steps in few-shot demonstrations. However, these methods do not experience firsthand the situations described by the language and the image. They

*Corresponding authors.

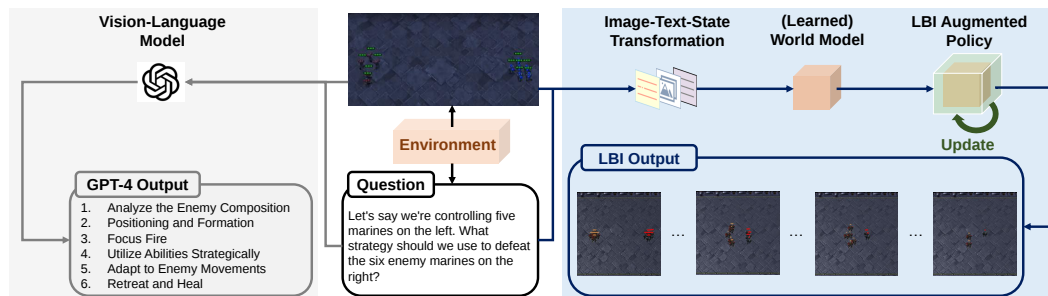


Figure 1: Complex decision problems that require a good understanding of the environment’s dynamics and the objective are still challenging for current vision-language models, e.g., the answer elicited by GPT-4 is sketchy and misleading. Instead, Learning before Interaction (LBI) enables grounded reasoning by simulating the task in the given question. LBI utilizes the simulator to train a MARL policy and generate the answer by running the converged policy on the simulator.

cannot find the correct answers through trial and error as humans, which is necessary to ground reasoning on complicated problems and transfer learned knowledge to unfamiliar domains. For example, as shown in Figure 1, when asked a complex multi-agent decision problem, one of the most widely-used large language models, GPT4 - though achieving superhuman performance in many reasoning tasks - will generate sketchy and misleading answers.

To tackle this problem, we can utilize the generative models to understand the properties of the task that the user describes and simulate the effects of the actions. We can derive the answer with a highly realistic simulator by experiment-reasoning or training any machine intelligence from simulated experience. The origin of this idea can be traced back to Dyna architecture (Sutton, 1990) and has spawned a series of model-based reinforcement learning (MBRL) theories and methods (Janner et al., 2019; Kaiser et al., 2019; Lai et al., 2020). Inspired by this, Mind’s Eye (Liu et al., 2022) enables language models to perform reasoning conditioned on the simulation results by running the corresponding experiment on a computational physics engine named MuJoCo (Todorov et al., 2012). Mind’s Eye can boost reasoning performance in zero-shot and few-shot settings by infusing such physical knowledge into language models. However, it is particularly designed for physical reasoning rather than decision-making problems.

In contrast, UniSim (Yang et al., 2024) formulates the action-in-video-out framework as an observation prediction diffusion model conditioned on finite history. It shows that the simulator learned from broad data can generalize to the real world and bridge the sim-to-real gap. Genie (Bruce et al., 2024) enables users to act in the generated environments on a frame-by-frame basis, opening the path for training generalist agents of the future. Notably, most of the existing breakthroughs on learning in the imagined experience have been focusing on single-agent scenarios and leave the world model largely unstudied for multi-agent reinforcement learning (MARL) tasks - it is common in real-world applications that multiple agents are required to solve a task in a coordinated fashion.

The roadblocks to building a simulator for MARL tasks are twofold. First, MARL tasks involve multiple entities’ attributes, e.g., positions and roles, making it difficult to describe a state using only text. The text and image information can be brought together to enrich the inputs for the simulator, but such a dataset is limited in quantity. Second, the dynamics and reward models of the MARL environment are more intricate than the single-agent setting. Current approaches assume the reward is known in the dataset (Meng et al., 2023) or can be easily deduced by the frame information (Yang et al., 2024), which could be challenging for MARL methods due to the abundance of agents’ tactics and the compositional nature of their functionalities.

This work explores a paradigm that adds language-guided simulation to the MARL pipeline to make policy learning grounded within the learned world model. First, we propose new offline MARL datasets to provide paired state-image information for the StarCraft Multi-Agent Challenge (SMAC) environment by transforming the state in the trajectory to the corresponding image. We also designed a parser to convert each trajectory to a task description using natural language. Then, we pre-train a vector quantized variational autoencoder (VQ-VAE) (Van Den Oord et al., 2017) to generate discrete representations for each frame. The world model is formulated as an interactive simulator that consists

of a dynamics and a reward model. The dynamics model comprises an image tokenizer and a causal transformer to generate interaction transitions autoregressively. The reward model is a bidirectional transformer learned by maximizing the likelihood of trajectories in the expert demonstrations under the task description.

Given a decision-making problem by the user and an image from the environment, we store the simulated interaction trajectories into a replay buffer by running an off-policy MARL algorithm on the generated dynamics model. Then, we utilize the generated reward model to label the reward for each state-action pair based on the whole trajectory. We update the policy network according to the reward with a behavior-regularization term, which serves as the conservatism for out-of-distribution state-action pairs. We use the image sequence generated by the interaction of the dynamics model and the converged policy model as the answer to the decision-making problem.

We summarize the main contributions of this paper in three folds: (1) It proposes novel MARL datasets for SMAC, where a parser automatically generates the ground-truth image of a given state and task description. (2) It introduces Learning before Interaction (LBI), an interactive simulator that generates trial-and-error experiences and improves the answers for multi-agent decision-making problems. (3) The empirical results show that LBI outperforms various offline learning methods on training and unseen MARL tasks. The visualization also indicates that LBI can produce consistent imagined trajectories and explainable rewards for interaction states.

2 Background

Decentralized Partially Observable Markov Decision Process. A fully cooperative multi-agent task in the partially observable setting can be formulated as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) (Oliehoek & Amato, 2016), consisting of a tuple $G = \langle A, S, \Omega, O, U, P, r, \gamma \rangle$, where $a \in A \equiv \{1, \dots, n\}$ is a set of agents, S is a set of states, and Ω is a set of joint observations. At each time step, each agent obtains its observation $o \in \Omega$ based on the observation function $O(s, a) : S \times A \rightarrow \Omega$, and an action-observation history $\tau_a \in T \equiv (\Omega \times U)^*$. Each agent a chooses an action $u_a \in U$ by a stochastic policy $\pi_a(u_a | \tau_a) : T \times U \rightarrow [0, 1]$, which forms a joint action $\mathbf{u}$. It results in a joint reward $r(s, \mathbf{u})$ and a transit to the next state $s' \sim P(\cdot | s, \mathbf{u})$. The formal objective function is to find the joint policy π that maximizes a joint action-value function $Q^\pi(s_t, \mathbf{u}_t) = r(s_t, \mathbf{u}_t) + \gamma \mathbb{E}_{s'} [V^\pi(s')]$, where $V^\pi(s) = \mathbb{E} [\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, \pi]$, and $\gamma \in [0, 1]$ is a discounted factor.

Inverse Reinforcement Learning. Suppose we do not have access to the ground truth reward function but have demonstrations $\mathcal{D}$ provided by an expert policy π_E . Imitation learning aims to directly learn policies that behave similarly to these demonstrations, whereas inverse reinforcement learning (IRL) seeks to infer the underlying reward functions which induce the expert policies. The MaxEnt IRL framework (Ziebart et al., 2008) aims to recover a reward function that rationalizes the expert behaviors with the least commitment, denoted as $\text{IRL}(\pi_E)$:

$$\begin{aligned} \text{IRL}(\pi_E) &= \arg \max_r \mathbb{E}_{\pi_E} [r(s, u)] - \text{RL}(r) \\ \text{RL}(r) &= \max_{\pi} \mathcal{H}(\pi) + \mathbb{E}_{\pi} [r(s, u)] \end{aligned} \quad (1)$$

where $\mathcal{H}(\pi) = \mathbb{E}_{\pi} [-\log \pi(u|s)]$ is the entropy of current policy π . It looks for a reward function that assigns a high reward to the expert policy and a low reward to the current policy π while searching for the best policy for the reward function in the inner loop.

3 Methodology

We formulate an interaction simulator as a transition prediction model that, given some state of the world and descriptions of the task, can take some actions as input and produce the consequence of the actions in the form of images, states, and rewards. In this paper, we consider building such simulators for a multi-agent decision-making environment named StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al., 2019), known for its rich environments and high control complexity. More information about SMAC can be found in Appendix C.

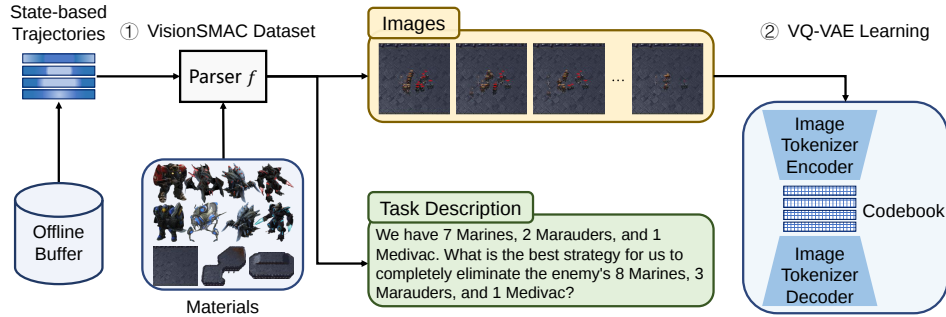


Figure 2: Datasets construction and VQ-VAE training.

3.1 VisionSMAC

The SMAC benchmark saves a replay of an episode as a SC2REPLAY file rather than providing the image feature during exploration. It is computationally expensive to construct datasets of images by watching such replay within the StarCraft II client and then subsampling a frame that captures meaningful actions. To solve this problem, we introduce VisionSMAC to convert the state into images and languages through a parser f , decoupled from StarCraft, making it easy to create new content.

First, we collect offline datasets across ten training maps in the SMAC benchmark by running multi-agent exploration methods named EMC (Zheng et al., 2021) and IIE (Liu et al., 2024). Each dataset contains a large number of interaction trajectories:

$$\tau := \{s_t, \{o_t^a\}_{a=1}^n, \{u_t^a\}_{a=1}^n, \{d_t^a\}_{a=1}^n\}_{t=0}^T \quad (2)$$

where s_t denotes the state, $\{o_t^a\}_{a=1}^n$ is the observation of each agent, $\{u_t^a\}_{a=1}^n$ is the joint action, and the done signal $d_t^a = 1$ when the agent a is killed at timestep t , n and T denote the number of agents and the length of the episode, respectively. We further collect the element images that appear in the game and affect the state, such as the units and the background terrain of training maps.

Given a multi-agent system and its interaction trajectory, the parser f reads predefined map information, such as the number and races of agents and enemies. Then, the parser converts the original state information into structured information, reading agents' and enemies' positions and health points. It will generate the corresponding interaction scenes by placing each unit image and its health bar at their positions with the corresponding background terrain. The image generated by the parser can resemble the frame subsampled from the original replay by running a SC2REPLAY file within the StarCraft II client, where the comparisons can be found in Appendix B. We also perform data augmentation to enable better feature abstraction by changing the background to different terrains.

Finally, we define a task description L to specify the environment and the task. The task description can be the terminated state, a slice of a trajectory, or any other representation of the episode. In this paper, we use the terrain information, the number and unit types of agents and enemies, and the sum of enemies' remaining health points at the terminated state as the task description. To this end, the parser reads the last state of the trajectory and extracts the remaining health points of both sides. We can obtain the practical task description by filling in predefined description templates (e.g., "Consider that we control {number of agents} {agent races} on the left.") and adding connecting words (e.g., "What plan should we use"). The detailed description of the datasets and the parser f can be found in Appendix B.

3.2 Training An Interactive Simulator

With trajectories with corresponding images and language guidance from different scenarios, we can formulate the interactions with StarCraft II as interacting with an interactive simulator. The simulator contains three key components: (1) an image tokenizer that converts raw video frame into discrete tokens, (2) a dynamics model that predicts the next frame and state given past frame and state tokens, (3) a reward model that infers the reward of a state-action pair given a trajectory. The idea behind decomposing the world model into a dynamics model and a reward model is to reuse the dynamics model for different tasks by combining it with any reward function.

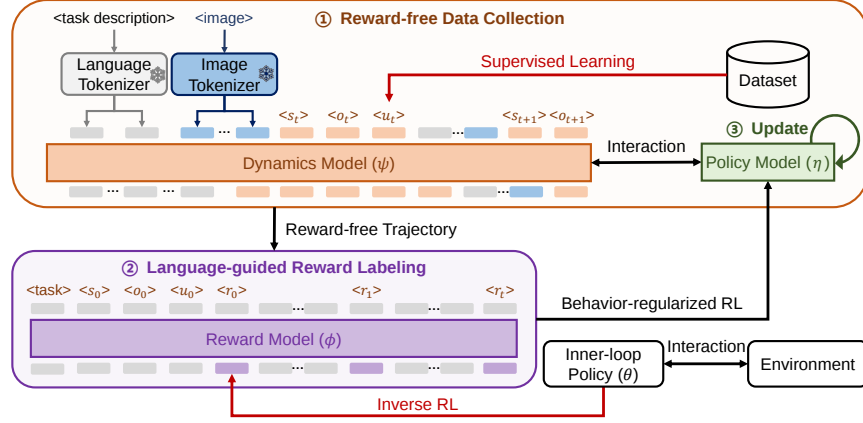


Figure 3: The overview of Learning before Interaction.

Image Tokenizer We compress images into discrete tokens to reduce dimensionality and enable higher-quality image generation. We make use of vector quantized variational autoencoder (VQ-VAE) (Van Den Oord et al., 2017), which takes every single image of the state as input, generating discrete representations. The tokenizer is trained using a standard VQ-VAE objective.

Dynamics Model The dynamics model is a causal transformer q parameterized by ψ , where the target sequence has the following form $x = \{\dots, L, z_t, s_t, o_t^1, \dots, o_t^n, u_t^1, \dots, u_t^n, z_{t+1}, s_{t+1}, \dots\}$, where z_t is the image representation generated by the fixed image tokenizer. We utilize the task description L to specify the dynamics of the environment, remaining consistent in one sequence. An embedding for each timestep is learned and added to each token. The dynamics model processes all past tokens and predicts future tokens via autoregressive modeling.

Then, we use the prediction heads to decode the predicted tokens to the corresponding element in the sequence and train them by minimizing the cross-entropy loss for actions and mean-squared error for others. The actions would serve as the reference policy for the learning with the simulated trajectories described in Section 3.3. In particular, we use a dynamics residual term to improve the accuracy and the stability of the generation by changing the target from s_{t+1} to $\Delta s_{t+1} = s_{t+1} - s_t$ for the state prediction head. We also apply this term to predict image representations. In addition, since the observation is only related to the current state and the vision range of the agents, we filter out the historical memories and use s_t as the input for the observation prediction.

Reward Model We resemble the training pipeline of inverse reinforcement learning - maximizing the likelihood of trajectories in the expert demonstrations while minimizing the likelihood of trajectories collected from an inner-loop policy. We introduce a reward function $\hat{r}$, which receives the entire trajectory as inputs to perform credit assignment under deterministic dynamics within the trajectory. We remake this formulation as a generalized version of the conventional reward design; if the reward function is Markovian, the temporal dependencies on other state-action pairs should always be zero.

To this end, we model the reward function as a bidirectional transformer model parameterized by ϕ , where the sequence is $\tilde{x} = \{\dots, s_t, L, u_t^1, \dots, u_t^n, \hat{r}_t, s_{t+1}, \dots\}$, and $\hat{r}_t = \{\hat{r}_t^a\}_{a=1}^N$ is a set of individual rewards for the agents. Again, we utilize the task description L to perform hindsight relabeling, which converts imperfect trajectories into possible solutions for reaching the last state s_T of the episode, generating the expert demonstration $\mathcal{D}$. We optimize the reward function by minimizing the following loss:

$$\nabla_{\phi} \mathcal{L} = -\mathbb{E}_{\tau \sim \mathcal{D}} \left[\sum_i \sum_t \gamma^t \nabla_{\phi} \hat{r}_t^a(\tau; \phi) \right] + \mathbb{E}_{\tau \sim \pi^{\theta}} \left[\sum_i \sum_t \gamma^t \nabla_{\phi} \hat{r}_t^a(\tau; \phi) \right], \quad (3)$$

where $\pi^{\theta} = \{\pi^a(u^a|s; \theta)\}_{a=1}^N$ is the inner-loop MA-SAC policy parameterized by θ , and we use the reward constraint by imposing an L2 penalization of the predicted rewards over all possible actions, which can be viewed as a conservative update for out-of-distribution state-action pairs. In particular, we alternate between k -step of policy update and k -step of reward update to avoid completely solving

the policy optimization subproblem before updating the reward parameters, where k is a given update iteration.

3.3 Inference: Learning Policy in the Simulator

We now describe how to generate grounded answers for multi-agent decision-making problems via LBI. Given an image of the initial state and a task description from the user, the agents interact with the dynamics model using a randomly initialized off-policy MARL algorithm to collect reward-free trajectories in an autoregressive manner. Then, the reward model predicts the immediate reward for each transition pair in the simulation trajectories. These relabeled trajectories are added to the replay buffer, serving as the training data for the policy network.

In practice, we construct the MARL problem in the simulator as a behavior-regularized MDP by imposing a behavior-regularization term:

$$\max_{\bar{\pi}} \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma \left(\sum_{a=1}^n \left(r_t^a(\tau; \phi) - \alpha \cdot \log \left(\frac{\bar{\pi}_i(u_t^a | o_t^a; \eta)}{q(u_t^a | x_{<u_t^a}; \psi)} \right) \right) \right) \right], \quad (4)$$

where $\bar{\pi} = \{\bar{\pi}_i(u_t^a | o_t^a; \eta)\}_{a=1}^n$ is the joint policy, and $q(u_t^a | x_{<u_t^a}; \psi)$ is the reference policy provided by the dynamics model. The last term enables in-sample learning and further mitigate the impact of exploring OOD regions in the state-action space. We use independent Q -learning to train the parameter-sharing agent network.

Since it is possible for specific agents to become inactive before the game terminates, we mark the terminated timestep for each agent and enemy once its predicted health is less than zero and then use zero vectors as the subsequent actions and observations. It can mitigate the hallucinating unrealistic outcomes - a dead agent performs a “moving” action. We also mask the predicted reward after the terminated timestep for the inactive agent to get a more accurate value estimate.

4 Related Work

World Models. There is a long-standing history of learning predictive models of the world. We list three categories of model-based reinforcement learning (MBRL) according to the type of model usage. The first category applies planning methods with world model simulation. AlphaGo (Silver et al., 2016) and MuZero (Schrittwieser et al., 2020) learn a transition model and apply Monte Carlo Tree Search to search for an action sequence with the highest accumulated rewards. By contrast, MBMF (Nagabandi et al., 2018), PETS (Chua et al., 2018), and PlaNet (Hafner et al., 2019b) integrate model predictive control (MPC) into the learned world model and sample high-reward action sequences. TD-MPC (Hansen et al., 2022) and TD-MPC2 (Hansen et al., 2024) utilize value functions to bootstrap the trajectories used for MPC planning.

The second category models a differentiable world model and utilizes the internal structure of the model to facilitate policy learning. GPS (Levine & Koltun, 2013) and GDP (Srinivas et al., 2018) perform differential planning and obtain the analytic form of the optimal policy. SVG (Heess et al., 2015) re-parameterizes the policy and the world model, then computes the policy gradient estimate by backpropagation via the world model. MAAC (Clavera et al., 2019), Dreamer (Hafner et al., 2019a) and its subsequent variants (Hafner et al., 2020, 2023) use the recurrent state-space model in PlaNet to learn the world model in a compact latent space and learn the policy entirely within this space.

The last one utilizes the learned world model to generate more experiences and then trains a policy on the dataset augmented by the model, also known as Dyna-style methods (Sutton, 1990). MVE (Feinberg et al., 2018) and STEVE (Buckman et al., 2018) depict a learned world model to calculate multi-step temporal-difference prediction for better value estimation. In contrast, SLBO (Luo et al., 2018), MBPO (Janner et al., 2019), and BMPO (Lai et al., 2020) theoretically analyze this learning framework and prove that the policy performance will improve monotonically in a world model with certain model bias and rollout length. To further increase the rollout length and avoid compounding errors, M2AC (Pan et al., 2020) and COPlanner (Wang et al., 2023) compute the uncertainty of each rollout step and perform conservative model rollouts by discarding the samples with high uncertainty or adding a penalty term into total reward. In practice, GAIA-1 (Hu et al., 2023), UniSim (Yang et al., 2024), and Genie (Bruce et al., 2024) show that the learned world model can enable the control policy to generalize to the real world when trained purely in simulation and bridge the sim-to-real

gap. These methods have impressive performance and theoretical bounds, attracting much research interest in the MBRL community. However, they focus on generating videos or trajectories that only involve one single agent instead of building a multi-agent simulator that can be used to further improve decision-making performance on coordination tasks in our work.

Imitation Learning. Imitation Learning (Bain & Sammut, 1995) formulates imitating an expert as a supervised learning problem, which has been widely adopted in various domains due to its simplicity and effectiveness (Silver et al., 2016; Swamy et al., 2020). GAIL (Ho & Ermon, 2016) and its extensions (Song et al., 2018; Ghasemipour et al., 2020) stand as a cornerstone approach, which trains a generator policy to imitate expert behaviors and a discriminator to distinguish between the expert and the learner’s state-action pair distributions. In light of the recent interest in foundational models, the conditional diffusion model is used to represent and learn an imitation learning policy, which produces a predicted action conditioning on a state and a sampled noise vector Pearce et al. (2022); Chi et al. (2023). These methods achieve encouraging results in modeling stochastic and multimodal behaviors from human experts or play data. DT-style methods (Chen et al., 2021; Wu et al., 2024) formulate the trajectory generation as a sequence modeling problem, which generates states, actions, and rewards by conditioning on a return-to-go token in an autoregressive manner.

In contrast, inverse reinforcement learning (IRL) is designed to infer the reward function that underlies the expert demonstrations, taking into account the temporal structure and showing better generalization than direct Behavioral Cloning (Ng & Russell, 2000; Ross et al., 2011; Barde et al., 2020). A main class of algorithms, Maximum entropy (MaxEnt) IRL (Haarnoja et al., 2017) and its extensions (Liu et al., 2021; Rolland et al., 2022), learn a stationary reward by minimizing divergence between the agent and expert distribution. Since the learned reward function can solve downstream tasks and transfer behavior across different dynamics, IRL is also helpful in several broader applications, e.g., IRL with natural language goals (Fu et al., 2018a; Zhou & Small, 2021; Xu et al., 2022), and RL with human feedback (Ziegler et al., 2019; Zhu et al., 2023; Wu et al., 2023), and dynamics learning (Luo et al., 2023). Furthermore, a series of sample-efficient algorithms are proposed to solve the MaxEnt IRL formulation (Fu et al., 2018b; Zeng et al., 2022, 2024). To side-step the expensive online environmental interactions in classic IRL, some work aims to learn a reward function from a static dataset by a variational Bayesian framework (Chan & van der Schaar, 2021), representing reward function via a learned soft Q -function (Garg et al., 2021), or incorporating conservatism into the estimated reward like offline Q -learning (Yue et al., 2022). The major bottleneck for these methods includes a lack of knowledge of the dynamics information and the reward overestimation for out-of-distribution state-action pairs. We formulate the reward model as a bidirectional transformer to receive the whole trajectory as the input, making it possible to solve non-Markovian rewards. In addition, we leverage the reward constraint and the behavior regularization to perform in-sample learning to avoid reward overestimation. The amount of expert demonstrations in these existing studies is also limited, as they do not treat hindsight relabeling via the textual description as an expert trajectory like in our work.

Offline Q -Learning. Offline Q -learning learns a policy from a fixed dataset where the reward is provided for each transition sample. Most off-policy reinforcement learning (RL) algorithms are applicable in offline Q -learning. However, they typically suffer from the overestimation problem of out-of-distribution (OOD) actions due to the distribution shift between the action distribution in the training dataset and that induced by the learned policy (Fujimoto et al., 2019). Several constraint methods are proposed to restrict the learned policy from producing OOD actions by leveraging importance sampling (Sutton et al., 2016; Nachum et al., 2019), incorporating explicit policy constraints (Kostrikov et al., 2021; Fakoor et al., 2021; Fujimoto & Gu, 2021; Tarasov et al., 2024), penalizing value estimates (Kumar et al., 2020; An et al., 2021; Shao et al., 2024), and uncertainty quantification (Wu et al., 2021; Zanette et al., 2021). Another branch resorts to learning without querying OOD actions and thus constrain the learning process within the support of the dataset (Bai et al., 2021; Lyu et al., 2022).

5 Experiments

In this section, we conduct empirical experiments to answer the following questions: (1) Is Learning before Interaction (LBI) better than the existing multi-agent reinforcement learning (MARL) methods in complex cooperative scenarios? (2) Can LBI generate long-horizon trajectories and reasonable reward functions at critical states? (3) Does LBI have the zero-shot ability to generalize to unseen

Table 1: Test win rates (%) and standard deviations compared with imitation learning methods.

Map Name	BC	MA-AIRL	MADT	MAPT	MA-TREX	LBI
1c3s5z	16.44± 1.35	7.88± 2.49	61.35± 7.26	74.77± 5.15	64.76± 11.62	94.59± 3.41
10m_vs_11m	26.19± 4.42	41.69± 7.12	82.76± 4.41	66.85± 9.28	48.78± 11.28	90.45± 6.99
2c_vs_64zg	17.37± 10.12	24.75± 10.83	61.90± 5.74	58.28± 7.84	22.45± 7.74	71.44± 8.83
3s_vs_5z	0.00± 0.00	0.05± 0.03	80.90± 0.45	72.33± 3.93	55.38± 18.03	92.82± 6.25
5m_vs_6m	13.78± 2.15	11.59± 6.75	79.78± 4.98	56.01± 3.17	50.01± 14.87	87.98± 5.10
6h_vs_8z	9.28± 5.06	16.47± 8.08	30.94± 25.54	37.16± 6.27	28.38± 5.31	66.61± 4.57
3s5z_vs_3s6z	0.00± 0.00	0.00± 0.00	27.44± 9.49	34.90± 6.84	36.16± 3.68	83.34± 4.27
corridor	0.00± 0.00	0.76± 0.15	69.85± 1.54	45.91± 15.47	30.59± 9.86	87.45± 2.94
MMM2	0.00± 0.00	0.00± 0.00	54.34± 12.83	19.21± 5.59	21.52± 6.58	95.96± 4.65

Table 2: Test return and standard deviations compared with offline reinforcement learning methods.

Map Name	BCQ-MA	CQL-MA	ICQ	OMAR	OMIGA	LBI
5m_vs_6m	9.13± 0.21	10.15± 0.15	9.47± 0.45	8.76± 0.52	10.38± 0.50	18.96± 0.56
2c_vs_64zg	18.86± 0.35	19.20± 1.25	18.47± 0.25	17.10± 0.94	19.25± 0.38	20.45± 0.25
6h_vs_8z	11.91± 0.44	9.95± 0.32	11.55± 0.15	9.74± 0.28	12.74± 0.21	18.97± 0.28
corridor	16.42± 1.55	6.64± 0.90	16.74± 1.78	8.15± 0.89	17.10± 1.33	19.50± 0.73

tasks? Then, we investigate the contribution of each component in the dynamics and the reward model. We provide the information of training datasets and experimental settings in Appendix B and D. We also discuss this paper’s broader impacts and limitations in Appendix A.1 and A.2.

5.1 Performance Comparison

Reward-free Offline Learning We compare LBI with the following imitation learning baselines: (1) BC: behavior cloning that imitates the whole datasets, (2) MA-AIRL (Yu et al., 2019): using adversarial learning to perform policy imitation, (3) MADT (Meng et al., 2023): utilizing the Decision Transformer (Chen et al., 2021) to perform sequence modeling, (4) MA-TREX: inferring the reward according to ranked demonstrations, the multi-agent version of TREX (Brown et al., 2019), (5) MAPT (Zhu et al., 2024): inferring the team rewards according to the preference return from a well-trained scripted teacher.

As shown in Table 1, LBI outperforms the baselines by a significant margin on various maps with different difficulty levels, indicating the importance and effectiveness of learning reward functions via the proposed world model. In contrast, BC and MA-AIRL fail to achieve success rates in most tasks because they imitate all past interaction sequences and cannot generalize and avoid sub-optimal solutions. MA-TREX and MAPT have plateaued in performance because they use the accumulated rewards and the preference deduced by the scripted teacher to specify the quality of the training data, respectively. MADT performs better than other baselines because Decision Transformer can be thought of as performing imitation learning on a subset of the data with a certain return.

Offline MARL We also compare LBI with the existing offline MARL methods with ground-truth rewards from the StarCraft Multi-Agent Challenge (SMAC), including the multi-agent version of BCQ (Fujimoto et al., 2019) and CQL (Kumar et al., 2020) (namely BCQ-MA and CQL-MA), ICQ (Yang et al., 2021), OMAR (Pan et al., 2022), and OMIGA (Wang et al., 2024). Table 2 shows that the performance of these offline MARL methods degrades dramatically with an increasing number of agents and is much lower than that of LBI. We hypothesize that the reasons for this gap are: (1) It is challenging and unnecessary to recover the Q -value based on the reward functions provided by SMAC (the hit-point damage dealt) because such reward design is inefficient for learning optimal policy. (2) These methods may introduce too much conservatism and affect the learning of the optimal policy, as the conservative update of the out-of-distribution (OOD) suboptimal policy that consists of some agents taking non-optimal actions and others taking optimal will inhibit the learning of the agents that take the optimal actions.

5.2 Generalization on Unseen Tasks

Since zero-shot generalization ability is crucial for generating grounded answers for multi-agent decision-making problems, we also test LBI’s ability to generalize to extensive unseen scenarios

Table 3: Test win rates (%) and standard deviations on unseen tasks.

Unseen Task	MADT	MA-TREX	LBI	Unseen Task	MADT	MA-TREX	LBI
1c3s	16.21 ± 5.38	23.53 ± 8.83	56.47 ± 5.63	1c2s7z	6.16 ± 3.09	5.69 ± 3.81	28.26 ± 6.41
6m	49.28 ± 4.06	37.12 ± 2.59	97.85 ± 2.15	6m_vs_7m	73.45 ± 7.22	32.88 ± 4.47	81.07 ± 5.17
1c_vs_32zg	2.08 ± 1.51	11.41 ± 3.41	58.33 ± 6.44	3s4z	90.21 ± 1.82	79.71 ± 3.56	87.55 ± 1.76
3s2z_vs_2s3z	0.00 ± 0.00	9.16 ± 5.62	18.22 ± 2.46	3s5z_vs_3s7z	10.21 ± 3.66	15.88 ± 4.34	22.08 ± 7.63
1c3s6z	16.41 ± 6.44	58.09 ± 3.41	65.38 ± 5.12	9m_vs_11m	76.44 ± 4.17	70.91 ± 6.95	75.05 ± 2.16

Table 4: The ablation results for the dynamics model without residual term (wo-RT), image reference (wo-IR), and using ground-truth image (GTI) as the reference for state prediction.

Algorithm	Prediction Error	Return (All)
LBI	0.016 ± 0.023	18.91 ± 1.33
LBI-GTI	0.014 ± 0.018	18.98 ± 0.89
LBI-wo-RT	0.434 ± 0.351	14.25 ± 1.84
LBI-wo-IR	0.029 ± 0.041	18.63 ± 1.01
LBI-wo-RT&IR	0.744 ± 1.164	12.13 ± 2.33

Table 5: The ablation results for the reward model without reward constraint (wo-RC), behavior regularization (wo-BR), and using ground-truth rewards (w-GTR) provided by the SMAC benchmark.

Algorithm	Return (Training)	Return (unseen)
LBI	19.47 ± 0.77	18.54 ± 1.49
LBI-GTR	16.68 ± 1.55	14.07 ± 2.79
LBI-wo-RC	17.85 ± 0.59	14.75 ± 1.67
LBI-wo-BR	18.82 ± 1.28	17.46 ± 2.01
LBI-wo-RC&BR	12.35 ± 2.38	9.83 ± 1.46

without retraining. Specifically, we evaluate our LBI and MADT on the ten unseen testing maps, varying agent numbers, action spaces, and levels of environment complexity. Table 3 shows that LBI consistently outperforms MADT in unseen scenarios by a large margin, successfully transferring knowledge to new tasks without requiring additional fine-tuning. It highlights that learning a reward function has better zero-shot generalization performance than simple policy adaptation.

5.3 Ablation Studies

In this section, we conduct ablation studies to analyze the contributions of each component in the dynamics model and the reward model across five evaluation runs on four training maps (6h_vs_8z, 3s5z_vs_3s6z, corridor, and MMM2) and four unseen maps (3s5z_vs_3s7z, 1c3s7z, 3s4z, 1c_vs_32zg). We show the results of the dynamics model in Table 4. Using the dynamics residual term is necessary to reduce the prediction error of the subsequent states and obtain good performance across all training and unseen tasks. The image reference is not so effective, even if we use ground-truth images as the reference. However, since images are more powerful in representing some situations than language or state information, we believe that the image serves as another modality to correct the prediction of the state. We would leave it for future work.

We demonstrate the ablation results of the reward model in Table 5. Compared with LBI-wo-RC&BR, the reward constraint and behavior regularization term can improve the overall performance on the training tasks. However, LBI-wo-BR performs better than LBI-wo-RC on unseen tasks, suggesting that the conservatism for reward is more important than the policy when OOD state-action pairs exist. The poor performance of LBI-w-GTR indicates that learning rewards from conditioned demonstrations may be more accessible and valuable for policy updates than reconstructing the pre-defined rewards by human experts.

5.4 Visualization

This section evaluates the dynamics model as a long-horizon policy-conditioned predictive model. Figure 4 showcases examples of length-40 image trajectories generated by the dynamics model, including MMM2, 3s_vs_5z, and 5m_vs_6m. We do not observe conspicuous compounding errors as the single-step prediction model does, highlighting that LBI has consistency and long-horizon generation ability. In the case of 5m_vs_6m, we present the following frames after taking one of the possible actions, showing that LBI can also perform action-controllable generation.

We also investigate the reward prediction at a critical junction in the state-action space that can transit to various states and significantly influence the success rate on the 5m_vs_6m task. At the moment, the agents have to learn to micromanage leapfrogging to achieve good coordination. Specifically, Agent 1 has a low health point and must move backward to avoid the enemies focusing fire on it; otherwise, the enemies will eliminate Agent 1 immediately and weaken our scarce forces. In Figure 4,

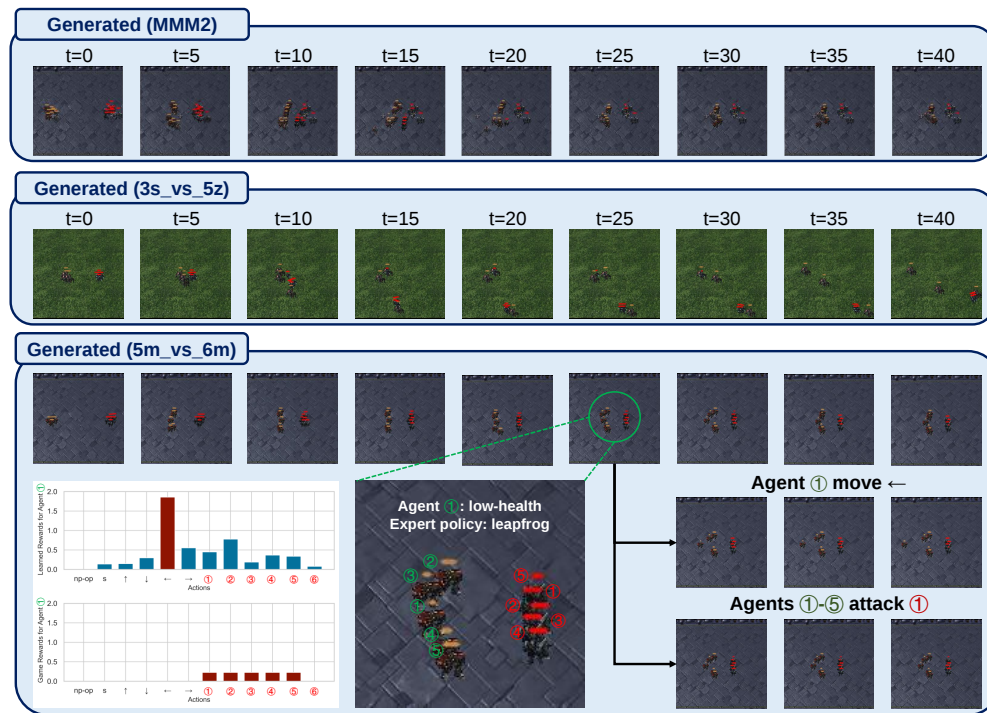


Figure 4: Visualization of the prediction from dynamics and reward model, where “np-op” and “s” denote no-operation and stopping, respectively.

we visualize the learned reward function of Agent 1, where the action space is no-operation, stopping, moving in cardinal directions, and selecting an enemy’s identity to attack. The learned reward for moving to the left is much higher than the other actions, allowing one to learn the optimal joint policy quickly. The rewards provided by the SMAC benchmark do not show this property, where multiple Monte Carlo samples are required to find the correct policy by estimating the expected return.

6 Conclusion and Future Work

We proposed Learning before Interaction (LBI), a novel paradigm that enables generative models to ground their answers for multi-agent decision-making problems with simulations between the world and the multi-agent system. We formulate an interactive simulator consisting of dynamics and reward models, given some states of the world and the task descriptions, generating the consequence of the actions in the form of images, states, and rewards. We hope the idea of including simulations in the reasoning will instigate broad interest in applying generative models to aid machine intelligence and decision-making.

7 Acknowledgements

This work was supported in part by NSFC under grant No.62125305, No.62088102, No. U23A20339, No. 62203348.

References

- Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.
- Chenjia Bai, Lingxiao Wang, Zhuoran Yang, Zhi-Hong Deng, Animesh Garg, Peng Liu, and Zhaoran Wang. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. In *International Conference on Learning Representations*, 2021.
- Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15*, pp. 103–129, 1995.
- Paul Barde, Julien Roy, Wonseok Jeon, Joelle Pineau, Chris Pal, and Derek Nowrouzezahrai. Adversarial soft advantage fitting: Imitation learning without policy optimization. *Advances in Neural Information Processing Systems*, 33:12334–12344, 2020.
- Wendelin Böhmer, Vitaly Kurin, and Shimon Whiteson. Deep coordination graphs. In *International Conference on Machine Learning*, pp. 980–991. PMLR, 2020.
- Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023a.
- Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on robot learning*, pp. 287–318. PMLR, 2023b.
- Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International conference on machine learning*, pp. 783–792. PMLR, 2019.
- Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. *arXiv preprint arXiv:2402.15391*, 2024.
- Jacob Buckman, Danijar Hafner, George Tucker, Eugene Brevdo, and Honglak Lee. Sample-efficient reinforcement learning with stochastic ensemble value expansion. *Advances in neural information processing systems*, 31, 2018.
- Alex J Chan and M van der Schaar. Scalable bayesian inverse reinforcement learning. In *International Conference on Learning Representations*, 2021.
- Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.
- Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.
- Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.
- Ignasi Clavera, Yao Fu, and Pieter Abbeel. Model-augmented actor-critic: Backpropagating through paths. In *International Conference on Learning Representations*, 2019.
- Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. In *International Conference on Machine Learning*, pp. 1538–1546. PMLR, 2019.
- Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*, pp. 8469–8488. PMLR, 2023.
- Rasool Fakoor, Jonas W Mueller, Kavosh Asadi, Pratik Chaudhari, and Alexander J Smola. Continuous doubly constrained batch reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 11260–11273, 2021.

- Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I Jordan, Joseph E Gonzalez, and Sergey Levine. Model-based value estimation for efficient model-free reinforcement learning. *arXiv preprint arXiv:1803.00101*, 2018.
- Justin Fu, Anoop Korattikara, Sergey Levine, and Sergio Guadarrama. From language to goals: Inverse reinforcement learning for vision-based instruction following. In *International Conference on Learning Representations*, 2018a.
- Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. In *International Conference on Learning Representations*, 2018b.
- Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.
- Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.
- Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems*, 34:4028–4039, 2021.
- Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In *Conference on robot learning*, pp. 1259–1277. PMLR, 2020.
- Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.
- Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2019a.
- 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*, pp. 2555–2565. PMLR, 2019b.
- Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2020.
- Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.
- Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations*, 2024.
- Nicklas A Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *International Conference on Machine Learning*, pp. 8387–8406. PMLR, 2022.
- Nicolas Heess, Gregory Wayne, David Silver, Timothy Lillicrap, Tom Erez, and Yuval Tassa. Learning continuous control policies by stochastic value gradients. *Advances in neural information processing systems*, 28, 2015.
- Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.
- Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.
- 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.
- Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model based reinforcement learning for atari. In *International Conference on Learning Representations*, 2019.
- Woojun Kim, Jongeui Park, and Youngchul Sung. Communication in multi-agent reinforcement learning: Intention sharing. In *International Conference on Learning Representations*, 2020.
- Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

- Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2021.
- Landon Kraemer and Bikramjit Banerjee. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190:82–94, 2016.
- Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. *arXiv preprint arXiv:2109.11251*, 2021.
- Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.
- Hang Lai, Jian Shen, Weinan Zhang, and Yong Yu. Bidirectional model-based policy optimization. In *International conference on machine learning*, pp. 5618–5627. PMLR, 2020.
- Martin Lauer. An algorithm for distributed reinforcement learning in cooperative multiagent systems. In *Proc. 17th International Conf. on Machine Learning*, 2000.
- Kuang-Huei Lee, Ofir Nachum, Mengjiao Sherry Yang, Lisa Lee, Daniel Freeman, Sergio Guadarrama, Ian Fischer, Winnie Xu, Eric Jang, Henryk Michalewski, et al. Multi-game decision transformers. *Advances in Neural Information Processing Systems*, 35:27921–27936, 2022.
- Sergey Levine and Vladlen Koltun. Guided policy search. In *International conference on machine learning*, pp. 1–9. PMLR, 2013.
- Minghuan Liu, Tairan He, Minkai Xu, and Weinan Zhang. Energy-based imitation learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 809–817, 2021.
- Ruibo Liu, Jason Wei, Shixiang Shane Gu, Te-Yen Wu, Soroush Vosoughi, Claire Cui, Denny Zhou, and Andrew M Dai. Mind’s eye: Grounded language model reasoning through simulation. In *The Eleventh International Conference on Learning Representations*, 2022.
- Yong Liu, Weixun Wang, Yujing Hu, Jianye Hao, Xingguo Chen, and Yang Gao. Multi-agent game abstraction via graph attention neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.
- Zeyang Liu, Lipeng Wan, Xue Sui, Zhuoran Chen, Kewu Sun, and Xuguang Lan. Deep hierarchical communication graph in multi-agent reinforcement learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 208–216, 2023.
- Zeyang Liu, Lipeng Wan, Xinrui Yang, Zhuoran Chen, Xingyu Chen, and Xuguang Lan. Imagine, initialize, and explore: An effective exploration method in multi-agent reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17487–17495, Mar. 2024. doi: 10.1609/aaai.v38i16.29698.
- Fan-Ming Luo, Tian Xu, Xingchen Cao, and Yang Yu. Reward-consistent dynamics models are strongly generalizable for offline reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2023.
- Yuping Luo, Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. In *International Conference on Learning Representations*, 2018.
- Jiafei Lyu, Xiaoteng Ma, Xiu Li, and Zongqing Lu. Mildly conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:1711–1724, 2022.
- Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. *Advances in Neural Information Processing Systems*, 32, 2019.
- Anuj Mahajan, Mikayel Samvelyan, Lei Mao, Viktor Makoviychuk, Animesh Garg, Jean Kossaifi, Shimon Whiteson, Yuke Zhu, and Animashree Anandkumar. Tesseract: Tensorised actors for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 7301–7312. PMLR, 2021.
- Laëtitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 64–69. IEEE, 2007.
- Linghui Meng, Muning Wen, Chenyang Le, Xiyun Li, Dengpeng Xing, Weinan Zhang, Ying Wen, Haifeng Zhang, Jun Wang, Yaodong Yang, et al. Offline pre-trained multi-agent decision transformer. *Machine Intelligence Research*, 20(2):233–248, 2023.

- Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in neural information processing systems*, 32, 2019.
- Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 7559–7566. IEEE, 2018.
- Andrew Y Ng and Stuart J Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 663–670, 2000.
- Frans A Oliehoek and Christopher Amato. *A concise introduction to decentralized POMDPs*. Springer, 2016.
- Frans A Oliehoek, Matthijs TJ Spaan, and Nikos Vlassis. Optimal and approximate q-value functions for decentralized pomdps. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.
- Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.
- Feiyang Pan, Jia He, Dandan Tu, and Qing He. Trust the model when it is confident: Masked model-based actor-critic. *Advances in neural information processing systems*, 33:10537–10546, 2020.
- Ling Pan, Longbo Huang, Tengyu Ma, and Huazhe Xu. Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification. In *International conference on machine learning*, pp. 17221–17237. PMLR, 2022.
- Emilio Parisotto and Russ Salakhutdinov. Efficient transformers in reinforcement learning using actor-learner distillation. In *International Conference on Learning Representations*, 2021.
- Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, et al. Stabilizing transformers for reinforcement learning. In *International conference on machine learning*, pp. 7487–7498. PMLR, 2020.
- Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, et al. Imitating human behaviour with diffusion models. In *The Eleventh International Conference on Learning Representations*, 2022.
- Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4295–4304. PMLR, 2018.
- Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:10199–10210, 2020.
- Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.
- Paul Rolland, Luca Viano, Norman Schürhoff, Boris Nikolov, and Volkan Cevher. Identifiability and generalizability from multiple experts in inverse reinforcement learning. *Advances in Neural Information Processing Systems*, 35:550–564, 2022.
- Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.
- Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.
- Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

- Jianzhun Shao, Yun Qu, Chen Chen, Hongchang Zhang, and Xiangyang Ji. Counterfactual conservative q learning for offline multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.
- David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.
- Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 5887–5896. PMLR, 2019.
- Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. Multi-agent generative adversarial imitation learning. *Advances in neural information processing systems*, 31, 2018.
- Aravind Srinivas, Allan Jabri, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Universal planning networks: Learning generalizable representations for visuomotor control. In *International conference on machine learning*, pp. 4732–4741. PMLR, 2018.
- Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2085–2087, 2018.
- Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pp. 216–224. Elsevier, 1990.
- Richard S Sutton, A Rupam Mahmood, and Martha White. An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of Machine Learning Research*, 17(73):1–29, 2016.
- Gokul Swamy, Siddharth Reddy, Sergey Levine, and Anca D Dragan. Scaled autonomy: Enabling human operators to control robot fleets. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5942–5948. IEEE, 2020.
- Yu Takagi and Shinji Nishimoto. High-resolution image reconstruction with latent diffusion models from human brain activity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14453–14463, 2023.
- Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.
- Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.
- Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.
- Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations*, 2020a.
- Tonghan Wang, Jianhao Wang, Chongyi Zheng, and Chongjie Zhang. Learning nearly decomposable value functions via communication minimization. In *International Conference on Learning Representations*, 2019.
- Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Roma: Multi-agent reinforcement learning with emergent roles. *arXiv preprint arXiv:2003.08039*, 2020b.
- Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. Rode: Learning roles to decompose multi-agent tasks. *arXiv preprint arXiv:2010.01523*, 2020c.
- Xiangsen Wang, Haoran Xu, Yinan Zheng, and Xianyuan Zhan. Offline multi-agent reinforcement learning with implicit global-to-local value regularization. *Advances in Neural Information Processing Systems*, 36, 2024.
- Xiyao Wang, Ruijie Zheng, Yanchao Sun, Ruonan Jia, Wichayaporn Wongkamjan, Huazhe Xu, and Fulong Huang. Coplanner: Plan to roll out conservatively but to explore optimistically for model-based rl. In *The Twelfth International Conference on Learning Representations*, 2023.
- Ermo Wei and Sean Luke. Lenient learning in independent-learner stochastic cooperative games. *The Journal of Machine Learning Research*, 17(1):2914–2955, 2016.

- Muning Wen, Jakub Grudzien Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. *arXiv preprint arXiv:2205.14953*, 2022.
- Yue Wu, Shuangfei Zhai, Nitish Srivastava, Joshua M Susskind, Jian Zhang, Ruslan Salakhutdinov, and Hanlin Goh. Uncertainty weighted actor-critic for offline reinforcement learning. In *International Conference on Machine Learning*, pp. 11319–11328. PMLR, 2021.
- Yueh-Hua Wu, Xiaolong Wang, and Masashi Hamaya. Elastic decision transformer. *Advances in Neural Information Processing Systems*, 36, 2024.
- Zequ Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 59008–59033. Curran Associates, Inc., 2023.
- Shusheng Xu, Huaijie Wang, and Yi Wu. Grounded reinforcement learning: Learning to win the game under human commands. *Advances in Neural Information Processing Systems*, 35:7504–7519, 2022.
- Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations*, 2024.
- Yaodong Yang, Jianye Hao, Guangyong Chen, Hongyao Tang, Yingfeng Chen, Yujing Hu, Changjie Fan, and Zhongyu Wei. Q-value path decomposition for deep multiagent reinforcement learning. In *International Conference on Machine Learning*, pp. 10706–10715. PMLR, 2020a.
- Yaodong Yang, Jianye Hao, Ben Liao, Kun Shao, Guangyong Chen, Wulong Liu, and Hongyao Tang. Qatten: A general framework for cooperative multiagent reinforcement learning. *arXiv preprint arXiv:2002.03939*, 2020b.
- Yiqin Yang, Xiaoteng Ma, Chenghao Li, Zewu Zheng, Qiyuan Zhang, Gao Huang, Jun Yang, and Qianchuan Zhao. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:10299–10312, 2021.
- Chao Yu, Akash Velu, Eugene Vinitzky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624, 2022.
- Lantao Yu, Jiaming Song, and Stefano Ermon. Multi-agent adversarial inverse reinforcement learning. In *International Conference on Machine Learning*, pp. 7194–7201. PMLR, 2019.
- Sheng Yue, Guanbo Wang, Wei Shao, Zhaofeng Zhang, Sen Lin, Ju Ren, and Junshan Zhang. Clare: Conservative model-based reward learning for offline inverse reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2022.
- Andrea Zanette, Martin J Wainwright, and Emma Brunskill. Provable benefits of actor-critic methods for offline reinforcement learning. *Advances in neural information processing systems*, 34:13626–13640, 2021.
- Siliang Zeng, Chenliang Li, Alfredo Garcia, and Mingyi Hong. Maximum-likelihood inverse reinforcement learning with finite-time guarantees. *Advances in Neural Information Processing Systems*, 35:10122–10135, 2022.
- Siliang Zeng, Chenliang Li, Alfredo Garcia, and Mingyi Hong. When demonstrations meet generative world models: A maximum likelihood framework for offline inverse reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.
- Lulu Zheng, Jiarui Chen, Jianhao Wang, Jiamin He, Yujing Hu, Yingfeng Chen, Changjie Fan, Yang Gao, and Chongjie Zhang. Episodic multi-agent reinforcement learning with curiosity-driven exploration. *Advances in Neural Information Processing Systems*, 34, 2021.
- Li Zhou and Kevin Small. Inverse reinforcement learning with natural language goals. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35-12, pp. 11116–11124, 2021.
- Banghua Zhu, Michael Jordan, and Jiantao Jiao. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *International Conference on Machine Learning*, pp. 43037–43067. PMLR, 2023.
- Tianchen Zhu, Yue Qiu, Haoyi Zhou, and Jianxin Li. Decoding global preferences: Temporal and cooperative dependency modeling in multi-agent preference-based reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38-15, pp. 17202–17210, 2024.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Broader Impacts and Limitations

A.1 Broader Impacts

Learning before Interaction provides grounded answers to complex multi-agent decision-making problems through the generation of simulators and trial-and-error learning. This can benefit those seeking to make decisions through long-term planning. With significant technological advancements, exploring the use of this technology may be crucial for enhancing existing human decision-making capabilities. For instance, negotiators could describe the opponent's personality traits and their decision-making limits to generate better negotiation strategies.

At the same time, we recognize that current generative simulators still cannot reliably generate state transitions across multiple domains, and learning joint multi-agent strategies still faces convergence difficulties. Therefore, Learning before Interaction may lead to incorrect decisions in specific fields. If humans intentionally follow the generated answers instead of using them as references, it could lead to unsafe or worse consequences. On the other hand, it could also have negative impacts when Learning before Interaction is misused in harmful applications if the generated environments and answers are sufficiently accurate.

A.2 Limitations

Although we have already seen significant improvements in reasoning capabilities for complex multi-agent tasks with Learning before Interaction, performance may be affected by the simulator's accuracy and the multi-agent policy learning performance. Unqualified simulators and difficult-to-converge multi-agent policies may lead to erroneous simulation results, which could be more misleading than the vague answers generated by existing visual language models. For example, the world model has limited out-of-domain generalization for domains that are not represented in the training data, e.g., unseen unit types. Further scaling up training data could help, as the parser can quickly and automatically generate images based on a given state.

While the learned reward functions can enhance the speed of multi-agent policy learning compared to other inverse reinforcement learning and online interaction learning methods, it still requires considerable waiting time to obtain a converged policy and the final answer. Such long waiting time is unacceptable in applications requiring real-time feedback, such as chatbots. One possible solution is to replace multi-agent reinforcement learning with planning methods based on the learned rewards and dynamics models, thereby accelerating the reasoning process. We will leave this issue in future work.

In addition, this paper is confined to scenarios within the game StarCraft II. This is an environment that, while complex, cannot represent the dynamics of all multi-agent tasks. Evaluation of multi-agent reinforcement learning algorithms, therefore, should not be limited to one benchmark but should target a variety with a range of tasks.

Map Name	Return Distribution	Map Name	Return Distribution
3s5z	19.43 ± 1.86	5m_vs_6m	19.83 ± 2.16
1c3s5z	19.66 ± 1.25	6h_vs_8z	18.84 ± 2.09
10m_vs_11m	19.75 ± 1.03	3s5z_vs_3s6z	19.76 ± 1.26
2c_vs_64zg	19.98 ± 0.71	corridor	19.69 ± 1.48
3s_vs_5z	19.88 ± 1.40	MMM2	19.63 ± 2.07

Table 6: Return distribution on training maps.

B Dataset Preparation

The training maps include 3s5z, 1c3s5z, 10m_vs_11m, 2c_vs_64zg, 3s_vs_5z, 5m_vs_6m, 6h_vs_8z, 3s5z_vs_3s6z, corridor, MMM2 in StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al., 2019). We use EMC (Zheng et al., 2021) and IIE (Liu et al., 2024) to collect 50000 trajectories for each map and save these data as NPY files. The data includes the states, the observations, the

terminated signals, the actions, the available actions, and the rewards. The return distribution on training maps is shown in Table 6. The average return is 19.64 ± 1.63 across ten training maps.

As shown in Figure 5, we collect the element images that appear in the game and affect the state, including units and background terrains of training maps. In Figure 6, we have presented the whole procedure of converting a state vector into an image for simulation and parsing a trajectory to produce a textual task description.

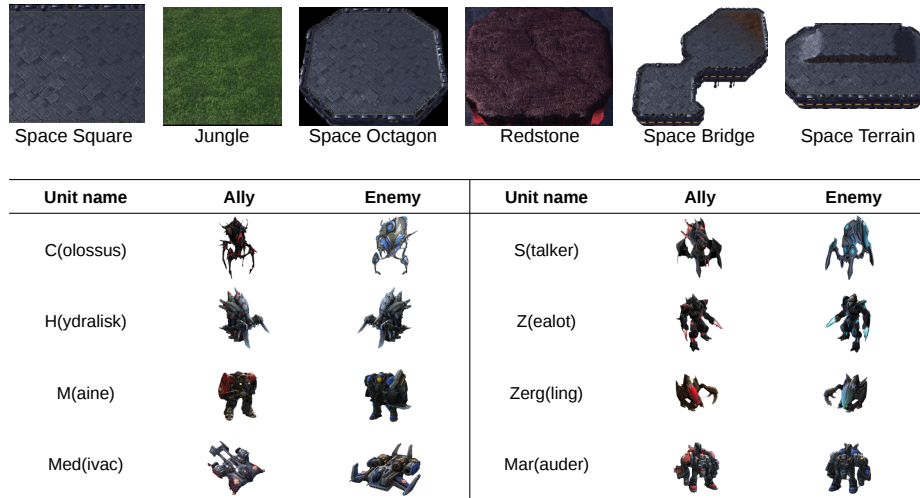


Figure 5: Images of units and terrains.



Figure 6: The whole pipeline of how the parser generates the image and the task description for a given state. Here, we only show three task descriptions the parser produces for demo purposes.

C StarCraft Multi-agent Challenge

StarCraft II is a real-time strategy game featuring three different races, Protoss, Terran, and Zerg, with different properties and associated strategies. The objective is to build an army powerful enough to destroy the enemy's base. When battling two armies, players must ensure army units are acting optimally. StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al., 2019) is a partially observable reinforcement learning benchmark built in StarCraft II. An individual agent with parameter sharing controls each allied unit, and a hand-coded built-in StarCraft II AI controls enemy units. The difficulty of the game AI is set to the "very difficult" level.

On the SMAC benchmark, agents can access their local observations within the field of view at each time step. The feature vector contains attributes of both allied and enemy units: `distance`, `relative_x`, `relative_y`, `health`, `shield`, and `unit_type`. In addition, agents can observe the last actions of allied units and the terrain features surrounding them. The global state vector includes the coordinates of all agents relative to the center of the map and other features present in the local observation of agents. The state stores the energy of Medivacs, the cooldown of the rest of the allied units, and the last actions of all agents. Note that the global state information is only available to agents during centralized training. All features in state and local observations are normalized by their maximum values. After receiving the observations, each agent is allowed to take action from a discrete set which consists of `move[direction]`, `attack[enemy_id]`, `stop` and `no-op`. Move direction includes north, south, east, and west. Note that the dead agents can only take `no-op` action while live agents cannot. For health units, Medivacs use `heal[agent_id]` actions instead of `attack[enemy_id]`.

Depending on different scenarios, the maximum number of actions varies between 7 and 70. Note that agents can only perform the `attack[enemy_id]` action when the enemy is within its shooting range. At each time step, agents take joint action and receive a positive global reward based on the total damage dealt to the enemy units. In addition, they can receive an extra reward of 10 points after killing each enemy unit and 200 points after killing all enemy units. The rewards are scaled to around 20, so the maximum cumulative reward is achievable in each scenario.

D Experiment Setting

In this section, we describe the ground-truth environment that agents interact, the implementation details of online learning methods, offline learning methods, and our model Learning before Interaction.

D.1 Online Learning

We adopt the same architectures for QMIX, QPLEX, CW-QMIX¹, RODE², MAVEN³, EMC⁴ as their official implementations (Samvelyan et al., 2019; Wang et al., 2020a; Rashid et al., 2020; Wang et al., 2020c; Mahajan et al., 2019; Zheng et al., 2021). Each agent independently learns a policy with fully shared parameters between all policies. We used RMSProp with a learning rate of $5e-4$ and $\gamma = 0.99$, buffer size 5000, and mini-batch size 32 for all algorithms. The dimension of each agent's GRU hidden state is set to 64.

For our experiments, we employ an ϵ -greedy exploration scheme for the joint policy, where ϵ decreases from 1 to 0.05 over 1 million timesteps in `6h_vs_8z`, `3s5z_vs_3s6z` and `corridor`, and over 50 thousand timesteps in other maps. The implementation of MAPPO is consistent with their official repositories⁵ (Yu et al., 2022). As shown in Table 7, all hyperparameters are left unchanged at the origin best-performing status. For CW-QMIX, the weight for negative samples is set to $\alpha = 0.5$ for all scenarios.

¹<https://github.com/oxwhirl/wqmixon>

²<https://github.com/TonghanWang/RODE>

³<https://github.com/AnujMahajanOxf/MAVEN>

⁴<https://github.com/kikojay/EMC>

⁵<https://github.com/zoeyuchao/mappo>

Hyperparameter	Value	Hyperparameter	Value
critic lr	5e-4	actor lr	5e-4
ppo epoch	5	ppo-clip	0.2
optimizer	Adam	batch size	3200
optim eps	1e-5	hidden layer	1
gain	0.01	training threads	32
rollout threads	8	γ	0.99
hidden layer dim	64	activation	ReLU

Table 7: Hyper-parameters in MAPPO.

All figures in online learning experiments are plotted using mean and standard deviation with confidence interval 95%. We conduct five independent runs with different random seeds for each learning curve.

D.2 Offline Learning

We adopt the same architectures for MA-AIRL⁶, MADT⁷, MAPT⁸, ICQ⁹, OMAR¹⁰, and OMIGA¹¹ as their official implementations (Yu et al., 2019; Meng et al., 2023; Zhu et al., 2024; Fujimoto et al., 2019; Kumar et al., 2020; Yang et al., 2021; Pan et al., 2022; Wang et al., 2024). We implement MA-TREX, BCQ-MA and CQL-MA based on TREX (Brown et al., 2019), BCQ (Fujimoto et al., 2019), and CQL (Kumar et al., 2020), respectively. In particular, we add the task description into MADT’s target sequence because it deprecates the reward-to-go term.

D.3 Learning before Interaction

We train our image tokenizer for 100k steps using the AdamW optimizer, with cosine decay, using the hyperparameters in Table 8. The batch size is 32, and the learning rate is 1e-4.

Component	Hyperparameter	Value
Encoder	num_layers	5e-4
	num_res_layers	2
	num_channels	(256,256)
	num_res_channels	(256,256)
	downsample	(2,4,1,1)
Decoder	num_layers	5e-4
	num_res_layers	2
	num_channels	(256,256)
	num_res_channels	(256,256)
	upsample	(2,4,1,1,0)
Codebook	num_codes	256
	latent_dim	32
	commitment_cost	0.25

Table 8: Hyper-parameters in VQ-VAE.

We build our dynamics model implementation based on Decision Transformer¹² (Chen et al., 2021). The complete list of hyperparameters can be found in Table 9. The dynamics models were trained using the AdamW optimizer.

⁶<https://github.com/ermongroup/MA-AIRL>

⁷<https://github.com/ReinholdM/Offline-Pre-trained-Multi-Agent-Decision-Transformer>

⁸<https://github.com/catezi/MAPT>

⁹<https://github.com/YiqinYang/ICQ>

¹⁰<https://github.com/ling-pan/OMAR>

¹¹<https://github.com/ZhengYinan-AIR/OMIGA>

¹²<https://github.com/kzl/decision-transformer>

Hyperparameter	Value	Hyperparameter	Value
number of layers	6	grad norm clip	1.0
attention heads	8	weight decay	0.1
embedding dims	64	Adam betas	(0.9,0.95)

Table 9: Hyperparameters in the transformer model.

The reward shares the same architecture as the dynamics model, but the attention mask in the transformer model is modified in order to receive the whole trajectory as input rather than the tokens that have come before the current one. Here are some tricks for reward learning: (1) we control the gap between the rewards of the expert behavior and the policy action - we stop the gradient for the reward of the expert behavior at a given state if it is greater than the one of the policy action, where β is the margin and set to 2; (2) we also set the target of unavailable actions' rewards to 0; (3) we alternate between k -step of policy update and reward update to avoid completely solving the policy optimization subproblem before updating the reward parameters, where $k = 5$.

In this paper, all experiments are implemented with Pytorch and executed on eight NVIDIA A800 GPUs.

E Additional Results

Using a Text-to-Code Converter can generate scenarios with the original game engine and then learn the joint policy. Therefore, we also consider the comparison with online MARL methods including CW-QMIX (Rashid et al., 2020), QPLEX (Wang et al., 2020a), MAVEN (Mahajan et al., 2019), EMC (Zheng et al., 2021), RODE (Wang et al., 2020c), QMIX (Rashid et al., 2018), MAPPO (Yu et al., 2022). Figure 7 demonstrates a significant improvement in the sample efficiency of LBI compared to the online MARL methods, suggesting that a pre-trained world model is necessary to reduce the waiting time for generating grounded answers for multi-agent decision-making problems.

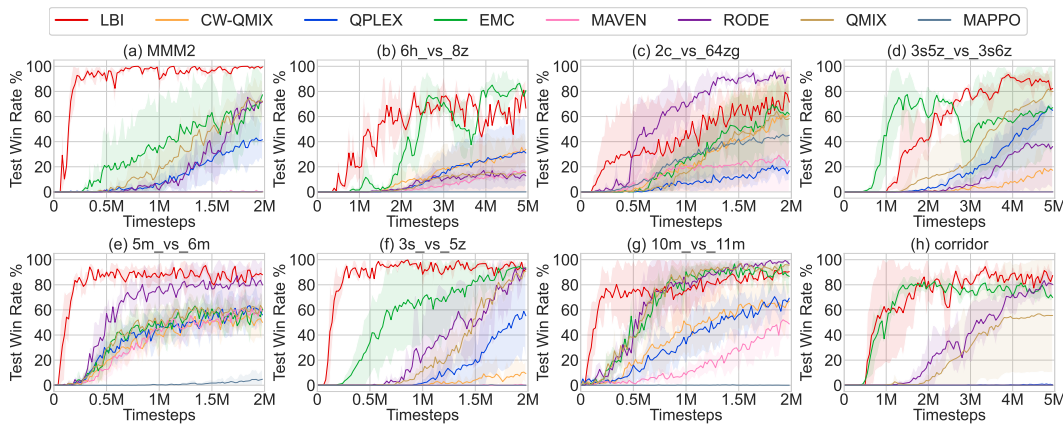


Figure 7: Performance comparisons between online learning methods using ground-truth rewards on the SMAC benchmark and LBI using the learned reward functions on the imagined world model.

In addition, we also show the qualitative comparison between the target and the generated sequences in Figure 8. Both trajectories are collected by running the same policy. We can see that the generated sequence can resemble the target one in most frames, but some differences exist in positions and health bars. However, compounding errors in the single-step model, which lead to physically implausible predictions, are not observed in the dynamics model generated by the causal transformer. For example, at the timestep of 10 in the MMM2 scenario, the generated frame does not contain the ally's Medivac, but we can see it in the following frames.

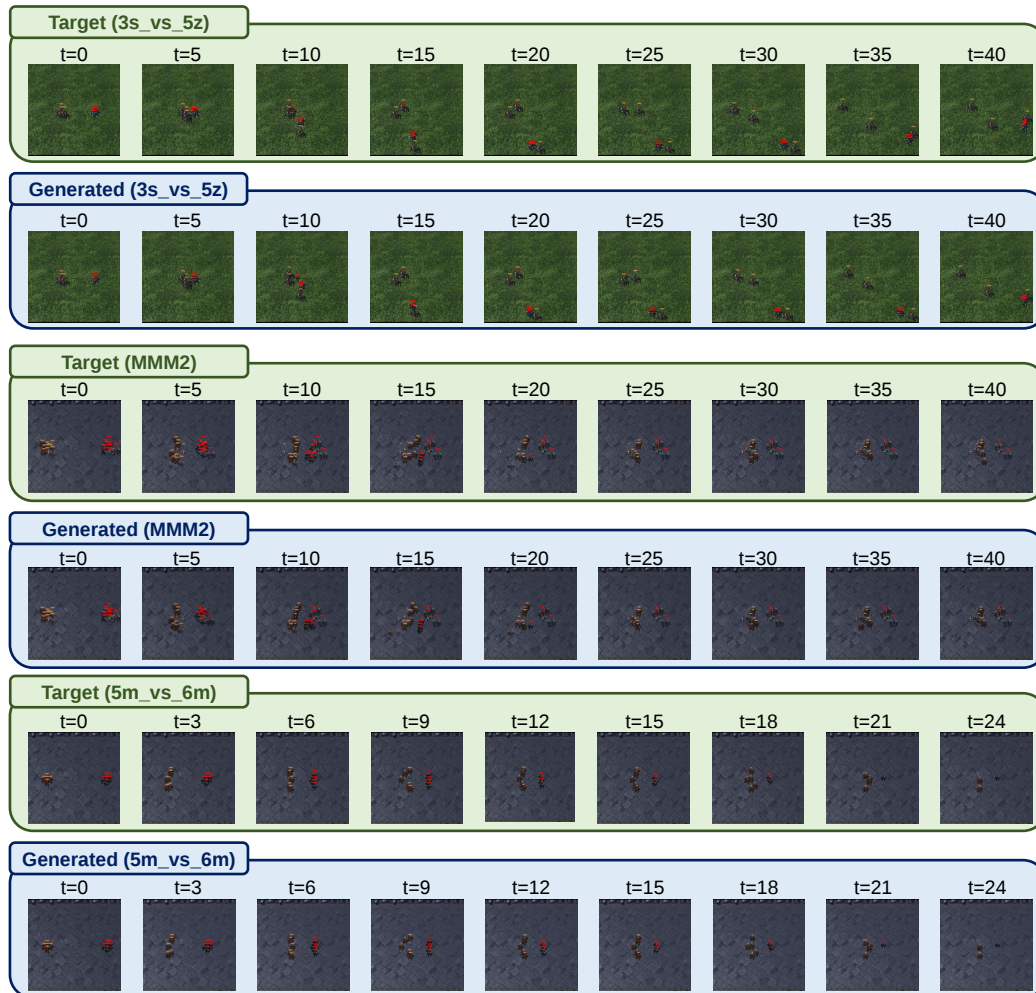


Figure 8: Comparisons of the target and the generated sequences across three different maps.

F Additional Related Work

Transformer Model Several works have explored the integration of transformer models into reinforcement learning (RL) settings. We classify them into two major categories depending on the usage pattern. The first category focuses on representing components in RL algorithms, such as policies and value functions (Parisotto et al., 2020; Parisotto & Salakhutdinov, 2021). These methods rely on standard RL algorithms to update policy, where the transformer only provides a large representation capacity and improves feature extraction. Conversely, the second category aims to replace the RL pipeline with sequence modeling. They autoregressively generate states, actions, and rewards by conditioning on the desired return-to-go during inference (Chen et al., 2021; Lee et al., 2022; Reed et al., 2022). Due to its simplicity and potential generalization ability, this category is widely used in various domains, such as robotics control (Brohan et al., 2023a; Padalkar et al., 2023; Driess et al., 2023) and multi-agent reinforcement learning (Meng et al., 2023; Liu et al., 2024).

Multi-agent Reinforcement Learning This section briefly introduces recent related work on cooperative multi-agent reinforcement learning (MARL). In the paradigm of centralized training with decentralized execution (CTDE), agents' policies are trained with access to global information in a centralized way and executed only based on local histories in a decentralized way (Oliehoek et al., 2008; Kraemer & Banerjee, 2016). One of the most significant challenges in CTDE is to ensure the correspondence between the individual Q -value functions and the joint Q -value function Q_{tot} , i.e., the Individual-Global Max (IGM) principle (Son et al., 2019). VDN (Sunehag et al., 2018)

and QMIX (Rashid et al., 2018) learn the joint Q -values and factorize them into individual Q -value functions in an additive and a monotonic fashion, respectively. Several works (Yang et al., 2020b,a; Wang et al., 2020b,c) have been proposed to improve the performance of QMIX, but as many previous studies pointed out, monotonic value function factorization limits the representational capacity of Q_{tot} and fails to learn the optimal policy when the target Q -value functions are non-monotonic (Mahajan et al., 2019; Son et al., 2019; Rashid et al., 2020). To solve this problem, some recent works (Wang et al., 2020a; Mahajan et al., 2021) try to achieve the full representational capacity of Q_{tot} , while others prioritize the potential optimal joint action and learn a biased Q_{tot} .

Some independent learning algorithms have also proven robust in solving multi-agent cooperative tasks. Distributed Q -learning (Lauer, 2000) and Hysteretic Q -learning (Matignon et al., 2007) place more importance on positive updates that increase a Q -value estimate, which is similar to the weighting function in WQMIX. However, Wei & Luke (2016) prove that these methods are vulnerable towards misleading stochasticity and propose LMRL2, where agents forgive the other's miscoordination in the initial exploration phase but become less lenient when the visitation of state-action pair increases. MAPPO (Yu et al., 2022) applies PPO (Schulman et al., 2017) into MARL and shows strong empirical performance. However, Kuba et al. (2021) points out MAPPO suffers from instability arising from the non-stationarity induced by simultaneously learning and exploring agents. Therefore, they introduce the sequential policy update scheme to achieve monotonic improvement on the joint policy.

Learning communication protocols to solve cooperative tasks is one of the desired emergent behaviors of agent interactions. It has recently become an active area in MARL, such as learning to share observations (Das et al., 2019; Wang et al., 2019; Liu et al., 2020) and intentions (Kim et al., 2020; Böhmer et al., 2020; Wen et al., 2022; Liu et al., 2023).

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 include the contributions made in the paper. See Section 1 for more information.

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: We discuss the limitations of this work in Appendix A.2, such as limited out-of-domain generalization and considerable cost time.

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: [NA]

Justification: NA.

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: We describe the steps taken to construct the dataset in Appendix B, and the implementation details of our model and baselines in Appendix D.

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 choose not to release the data and code at present. We would like to have the opportunity to further engage with the research community and to ensure that any future such releases are respectful, safe, and responsible.

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: We provide the training and test details in Appendix D.

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: We conduct five independent runs with different random seeds for each result. The results are accompanied by standard deviations.

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: We provide the information on the computer resources in Appendix D.

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: We conduct the research with 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: We discuss both potential positive societal impacts and negative societal impacts in Appendix A.1.

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: [\[Yes\]](#)

Justification: We does not use a pre-trained model (which may generate unsafe images), and we construct the image dataset through a parser. See Appendix B for more information.

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: We cite the original paper and provide the URLs for the assets in Appendix D.

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: We describe the steps taken to construct the dataset in Appendix B, and the implementation details of our model and baselines in Appendix D. However, We choose not to release the data and code at present. We would like to have the opportunity to further engage with the research community and to ensure that any future such releases are respectful, safe and responsible.

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: NA.

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: NA.

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.