# Relational Reasoning via Set Transformers: Provable Efficiency and Applications to MARL

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

The cooperative Multi-Agent Reinforcement Learning (MARL) with permutation invariant agents framework has achieved tremendous empirical successes in real-world applications. Unfortunately, the theoretical understanding of this MARL problem is lacking due to the curse of many agents and the limited exploration of the relational reasoning in existing works. In this paper, we verify that the transformer implements complex relational reasoning, and we propose and analyze model-free and model-based offline MARL algorithms with the transformer approximators. We prove that the suboptimality gaps of the model-free and model-based algorithms are independent of and logarithmic in the number of agents respectively, which mitigates the curse of many agents. These results are consequences of a novel generalization error bound of the transformer and a novel analysis of the Maximum Likelihood Estimate (MLE) of the system dynamics with the transformer. Our model-based algorithm is the first provably efficient MARL algorithm that explicitly exploits the permutation invariance of the agents.

## 1 Introduction

Cooperative MARL algorithms have achieved tremendous successes across a wide range of real-world applications including robotics [1, 2], games [3, 4], and finance [5]. In most of these works, the *permutation invariance* of the agents is embedded into the problem setup, and the successes of these works hinge on leveraging this property. However, the theoretical understanding of why the permutation invariant MARL has been so successful is lacking due to the following two reasons. First, the size of the state-action space grows exponentially with the number of agents; this is known as "the curse of many agents" [6, 7]. The exponentially large state-action space prohibits the learning of value functions and policies due to the curse of dimensionality. Second, although the mean-field approximation is widely adopted to mitigate the curse of many agents [6, 8], this approximation fails to capture the complex interplay between the agents. In the mean-field approximation, the influence of all the other agents on a fixed agent is captured only through the empirical distribution of the local states and/or local actions [6, 8]. This induces a restricted class of function approximators, which nullifies the possibly complicated relational structure of the agents, and thus fails to incorporate the complex interaction between agents. Therefore, designing provably efficient MARL algorithms that incorporate the efficient relational reasoning and break the curse of many agents remains an interesting and meaningful question.

In this paper, we take the transformer networks as the representation learning module to incorporate relational reasoning among the agents. In particular, we focus on the offline MARL problem with the transformer approximators in the cooperative setting. In this setting, all the agents learn policies *cooperatively* to maximize a common reward function. More specifically, in the offline setting, the learner only has access to a pre-collected dataset and cannot interact adaptively with the environment.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Moreover, we assume that the underlying Markov Decision Process (MDP) is *homogeneous*, which means that the reward and the transition kernel are permutation invariant functions of the state-action pairs of the agents. Our goal is to learn an *optimal* policy that is also permutation invariant.

To design provably efficient offline MARL algorithms, we need to overcome three key challenges. (i) To estimate the action-value function and the system dynamics, the approximator function needs to implement efficient relational reasoning among the agents. However, the theoretically-grounded function structure that incorporates the complex relational reasoning needs to be carefully designed. (ii) To mitigate the curse of many agents, the generalization bound of the transformer should be independent of the number of agents. Existing results in [9] thus require rethinking and improvements. (iii) In offline Reinforcement Learning (RL), the mismatch between the sampling and visitation distributions induced by the optimal policy (i.e., "distribution shift") greatly restricts the application of the offline RL algorithm. Existing works adopt the *"pessimism"* principle to mitigate such a challenge. However, this requires the quantification of the uncertainty in the value function estimation and the estimation of the dynamics in the model-free and model-based methods respectively. The quantification of the estimation error with the transformer function class is a key open question.

We organize our work by addressing the abovementioned three challenges.

First, we theoretically identify the function class that can implement complex relational reasoning. We demonstrate the relational reasoning ability of the attention mechanism by showing that approximating the self-attention structure with the permutation invariant fully-connected neural networks (i.e., deep sets [10]) requires an *exponentially large* number of hidden nodes in the input dimension of each channel (Theorem 1). This result necessitates the self-attention structure in the set transformer.

Second, we design offline model-free and model-based RL algorithms with the transformer approximators. In the former, the transformer is adopted to estimate the action-value function of the policy. The *pessimism* is encoded in that we learn the policy according to the *minimal* estimate of the action-value function in the set of functions with bounded empirical Bellman error. In the model-based algorithm, we estimate the system dynamics with the transformer structure. The policy is learned pessimistically according to the estimate of the system dynamics in the confidence region that induces the conservative value function.

Finally, we analyze the suboptimality gaps of our proposed algorithms, which indicate that the proposed algorithms mitigate the curse of many agents. For the model-free algorithm, the suboptimality gap in Theorem 2 is independent of the number of agents, which is a consequence of the fact that the generalization bound of the transformer (Proposition 2) is independent of the number of channels. For the model-based algorithm, the bound on the suboptimality gap in Theorem 3 is logarithmic in the number of agents; this follows from the analysis of the MLE of the system dynamics in Proposition 3. We emphasize that our model-based algorithm is the first provably efficient MARL algorithm that exploits the permutation equivariance when estimating the dynamics.

**Technical Novelties.** In Proposition 2, we leverage a PAC-Bayesian framework to derive a generalization error bound of the transformer. Compared to [9, Theorem 4.6], the result is a significant improvement in the dependence on the number of channels $N$ and the depth of neural network $L$. This result may be of independent interest for the theoretical understanding of the attention mechanism. In Proposition 3, we derive the first estimation uncertainty quantification of the system dynamics with the transformer approximators, which can be also be used to analyze other RL algorithms with such approximators.

**More Related Work.** In this paper, we consider the offline RL problem, and the insufficient coverage lies at the core of this problem. With the global coverage assumption, a number of works have been proposed from both the model-free [11–15] and model-based [11, 16] perspectives. To weaken the global coverage assumption, we leverage the "pessimism" principle in the algorithms: the model-free algorithms impose additional penalty terms on the estimate of the value function [17, 18] or regard the function that attains the minimum in the confidence region as the estimate of the value function [19]; the model-based algorithms estimate the system dynamics by incorporating additional penalty terms [20] or minimizing in the region around MLE [21]. For the MARL setting, the offline MARL with the mean-field approximation has been studied in [8, 22].

The analysis of the MARL algorithm with the transformer approximators requires the generalization bound of the transformer. The transformer is an element of the group equi/invariant functions, whose benefit in terms of its generalization capabilities has attracted extensive recent attention.

Generalization bounds have been successively improved by analyzing the cardinality of the "effective" input field and Lipschitz constants of functions [23, 24]. However, these methods result in loose generalization bounds when applied to deep neural networks [25]. Zhu, An, and Huang [26] empirically demonstrated the benefits of the invariance in the model by refining the covering number of the function class, but a unified theoretical understanding is still lacking. The covering number of the norm-bounded transformer was shown by [9] to be at most logarithmic in the number of channels. We show that this can be further improved using a PAC-Bayesian framework. In addition, we refer to the related concurrent work [27] for a Rademacher complexity based generalization bound of the transformer that is independent of the length of the sequence for the tasks such as computer vision.

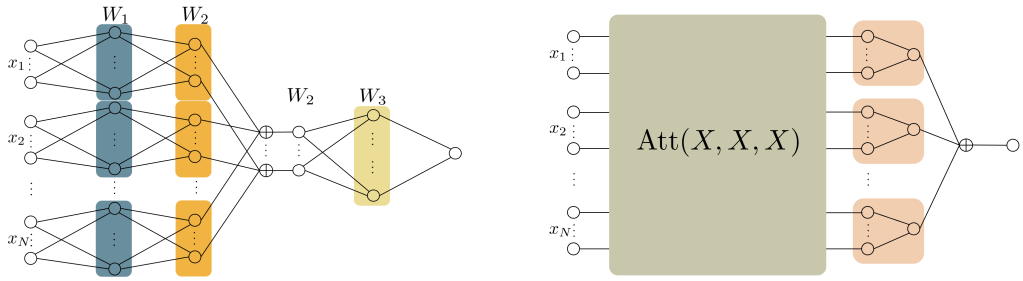
## 2 Preliminaries

**Notation.** Let $[n] = \{1, \ldots, n\}$. The $i^{\text{th}}$ entry of the vector $x$ is denoted as $x_i$ or $[x]_i$. The $i^{\text{th}}$ row and the $i^{\text{th}}$ column of matrix $X$ are denoted as $X_{i,:}$ and $X_{:,i}$ respectively. The $\ell_p$*-norm* of the vector $x$ is $\|x\|_p$. The $\ell_{p,q}$*-norm* of the matrix $X \in \mathbb{R}^{m\times n}$ is defined as $\|X\|_{p,q} = (\sum_{i=1}^{n} \|X_{:,i}\|_p^q)^{1/q}$, and the *Frobenius norm* of $X$ is defined as $\|X\|_{\mathrm{F}} = \|X\|_{2,2}$. The *total variation distance* between two distributions $P$ and $Q$ on $\mathcal{A}$ is defined as $\mathrm{TV}(P, Q) = \sup_{A\subseteq\mathcal{A}} |P(A) - Q(A)|$. For a set $\mathcal{X}$, we use $\Delta(\mathcal{X})$ to denote the set of distributions on $\mathcal{X}$. For two conditional distributions $P, Q : \mathcal{X} \to \Delta(\mathcal{Y})$, the $d_\infty$ distance between them is defined as $d_\infty(P, Q) = 2\sup_{x\in\mathcal{X}} \mathrm{TV}(P(\cdot\,|\,x), Q(\cdot\,|\,x))$. Given a metric space $(\mathcal{X}, \|\cdot\|)$, for a set $\mathcal{A} \subseteq \mathcal{X}$, an $\varepsilon$*-cover* of $\mathcal{A}$ is a finite set $\mathcal{C} \subseteq \mathcal{X}$ such that for any $a \in \mathcal{A}$, there exists $c \in \mathcal{C}$ and $\|c - a\| \le \varepsilon$. The $\varepsilon$*-covering number* of $\mathcal{A}$ is the cardinality of the smallest $\varepsilon$-cover, which is denoted as $\mathcal{N}(\mathcal{A}, \varepsilon, \|\cdot\|)$.

**Attention Mechanism and Transformers.** The *attention mechanism* is a technique that mimics cognitive attention to process multi-channel inputs [28]. Compared with the Convolutional Neural Network (CNN), the transformer has been empirically shown to possess outstanding robustness against occlusions and preserve the global context due to its special relational structure [29]. Assume we have $N$ query vectors that are in $\mathbb{R}^{d_Q}$. These vectors are stacked to form the matrix $Q \in \mathbb{R}^{N\times d_Q}$. With $N_V$ *key vectors* in the matrix $K \in \mathbb{R}^{N_V\times d_Q}$ and $N_V$ *value vectors* in the matrix $V \in \mathbb{R}^{N_V\times d_V}$, the attention mechanism maps the queries $Q$ using the function $\mathrm{Att}(Q, K, V) = \mathrm{SM}(QK^\top)V$, where $\mathrm{SM}(\cdot)$ is the row-wise softmax operator that normalizes each row using the exponential function, i.e., for $x \in \mathbb{R}^d$, $[\mathrm{SM}(x)]_i = \exp(x_i)/\sum_{j=1}^{d} \exp(x_j)$ for $i \in [d]$. The product $QK^\top$ measures the similarity between the queries and the keys, which is then passed through the activation function $\mathrm{SM}(\cdot)$. Thus, $\mathrm{SM}(QK^\top)V$ essentially outputs a weighted sum of $V$ where a value vector has greater weight if the corresponding query and key are more similar. The *self-attention mechanism* is defined as the attention that takes $Q = XW_Q$, $K = XW_K$ and $V = XW_V$ as inputs, where $X \in \mathbb{R}^{N\times d}$ is the input of the self-attention, and $W_Q, W_K \in \mathbb{R}^{d\times d_Q}$ and $W_V \in \mathbb{R}^{d\times d_V}$ are the parameters. Intuitively, the self-attention mechanism weighs the inputs with the correlations among the $N$ different channels. This mechanism demonstrates a special pattern of *relational reasoning* among the channels of $X$.

In addition, the self-attention mechanism is *permutation invariant* in the channels in $X$. This implies that for any row-wise permutation function $\psi(\cdot)$, which swaps the rows of the input matrix according to a given permutation of $[N]$, we have $\mathrm{Att}(\psi(X)W_Q, \psi(X)W_K, \psi(X)W_V) = \psi(\mathrm{Att}(XW_Q, XW_K, XW_V))$. The permutation equivariance of the self-attention renders it suitable for inference tasks where the output is equivariant with respect to the ordering of inputs. For example, in image segmentation, the result should be invariant to the permutation of the objects in the input image [30]. The resultant transformer structure combines the self-attention with multi-layer perceptrons and composes them to form deep neural networks. It remains permutation equi/invariant with respect to the order of the channels and has achieved excellent performance in many tasks [31–33].

**Offline Cooperative MARL.** In this paper, we consider the *cooperative* MARL problem, where all agents aim to maximize a *common* reward function. The corresponding MDP is characterized by the tuple $(\bar{S}_0, \bar{\mathcal{S}}, \bar{\mathcal{A}}, P^*, r, \gamma)$ and the number of agents is $N$. The state space $\bar{\mathcal{S}} = \mathcal{S}^N$ is the Cartesian product of the state spaces of each agent $\mathcal{S}$, and $\bar{S} = [s_1, \ldots, s_N]^\top$ is the state, where $s_i \in \mathbb{R}^{d_\mathcal{S}}$ is the state of the $i^{\text{th}}$ agent. The initial state is $\bar{S}_0$. The action space $\bar{\mathcal{A}} = \mathcal{A}^N$ is the Cartesian product of the action spaces $\mathcal{A}$ of each agent, and $\bar{A} = [a_1, \ldots, a_N]^\top$ is the action, where $a_i \in \mathbb{R}^{d_\mathcal{A}}$ is the action of the $i^{\text{th}}$ agent. The transition kernel is $P^* : \mathcal{S}^N \times \mathcal{A}^N \to \Delta(\mathcal{S}^N)$, and $\gamma \in (0, 1)$ is the *discount factor*. Without loss of generality, we assume that the reward function $r$




(a) $\rho_{\mathrm{ReLU}}(\sum_{i=1}^{N} \phi_{\mathrm{ReLU}}(x_i))$ with $\rho_{\mathrm{ReLU}}$ and $\psi_{\mathrm{ReLU}}$ as single-hidden layer neural networks.

(b) Self-attention mechanism $\mathbb{I}_N^\top \mathrm{Att}(\mathrm{X}, \mathrm{X}, \mathrm{X})\mathrm{w}$.

Figure 1: The blocks with the same color share the same parameters. The left figure shows that $\rho_{\mathrm{ReLU}}(\sum_{i=1}^{N} \phi_{\mathrm{ReLU}}(x_i))$ first sums the outputs of $\phi_{\mathrm{ReLU}}(x_i)$, and it implements the relational reasoning only through the single-hidden layer network $\rho_{\mathrm{ReLU}}$. In contrast, the self-attention block in the right figure captures the relationship among channels and then sums the outputs of each channel.

is deterministic and bounded, i.e., $r : \mathcal{S}^N \times \mathcal{A}^N \to [-R_{\max}, R_{\max}]$. We define the the *state-value function* $V_P^\pi : \mathcal{S}^N \to [-V_{\max}, V_{\max}]$, where $V_{\max} = R_{\max}/(1-\gamma)$, and the *action-value function* $Q_P^\pi : \mathcal{S}^N \times \mathcal{A}^N \to [-V_{\max}, V_{\max}]$ of a policy $\pi$ and a transition kernel $P$ as

$$V_P^\pi(\bar{S}) = \mathbb{E}^\pi\left[\sum_{t=0}^{\infty} \gamma^t r(\bar{S}_t, \bar{A}_t) \,\middle|\, \bar{S}_0 = \bar{S}\right] \text{ and } Q_P^\pi(\bar{S}, \bar{A}) = \mathbb{E}^\pi\left[\sum_{t=0}^{\infty} \gamma^t r(\bar{S}_t, \bar{A}_t) \,\middle|\, \bar{S}_0 = \bar{S}, \bar{A}_0 = \bar{A}\right],$$

respectively. Here, the expectation is taken with respect to the Markov process induced by the policy $\bar{A}_t \sim \pi(\cdot \mid \bar{S}_t)$ and the transition kernel $P$. The action-value function $Q_{P^*}^\pi$ is the unique fixed point of the operator $(\mathcal{T}^\pi f)(\bar{S}, \bar{A}) = r(\bar{S}, \bar{A}) + \gamma \mathbb{E}_{\bar{S}' \sim P^*(\cdot \mid \bar{S}, \bar{A})}[f(\bar{S}', \pi) \mid \bar{S}, \bar{A}]$, where the term in the expectation is defined as $f(\bar{S}, \pi) = \mathbb{E}_{\bar{A} \sim \pi(\cdot \mid \bar{S})}[f(\bar{S}, \bar{A})]$. We further define the *visitation measure* of the state and action pair induced the policy $\pi$ and transition kernel $P$ as $d_P^\pi(\bar{S}, \bar{A}) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t d_{P,t}^\pi$, where $d_{P,t}^\pi$ is the distribution of the state and the action at step $t$.

In offline RL, the learner only has access to a pre-collected dataset and cannot interact with the environment. The dataset $\mathcal{D} = \{(\bar{S}_i, \bar{A}_i, r_i, \bar{S}_i')\}_{i=1}^n$ is collected in an i.i.d. manner, i.e., $(\bar{S}_i, \bar{A}_i)$ is independently sampled from $\nu \in \Delta(\bar{\mathcal{S}} \times \bar{\mathcal{A}})$, and $\bar{S}_i' \sim P^*(\cdot \mid \bar{S}_i, \bar{A}_i)$. Given a policy class $\Pi$, our goal is to find an optimal policy that maximizes the state-value function $\pi^* = \mathrm{argmax}_{\pi \in \Pi} V_{P^*}^\pi(\bar{S}_0)$. For any $\pi \in \Pi$, the *suboptimality gap* of $\pi$ is defined as $V_{P^*}^{\pi^*}(\bar{S}_0) - V_{P^*}^\pi(\bar{S}_0)$.

## 3 How to Define a Proper Function Class for Multi-Agent RL with Relational Reasoning?

Before designing the RL algorithms, we first need to specify the function class that is adopted to estimate the action-value function and the system dynamics. We adopt the set transformer neural network, which combines the self-attention mechanism and the fully-connected neural network. Although the composition of the fully-connected layers in the deep sets manner also remains permutation invariant [10], we show that it loses the complex relational reasoning process in the set transformer unless its width is exponential in the input dimension.

We demonstrate this claim by approximating the self-attention mechanism with a permutation invariant fully-connected neural network. Zaheer et al. [10, Theorem 2] showed that all permutation invariant functions take the form $\rho(\sum_{i=1}^{N} \phi(x_i))$ with $X = [x_1, \ldots, x_N]^\top \in \mathbb{R}^{N \times d}$ as the input. Since the single-hidden layer ReLU neural network is an universal approximator for continuous functions [34], we set $\phi : \mathbb{R}^{N \times d} \to \mathbb{R}^{W_2}$ and $\rho : \mathbb{R}^{W_2} \to \mathbb{R}$ to be single-hidden layer neural networks with ReLU activation functions as shown in Figure 1(a), where $W_2$ is the dimension of the intermediate outputs. The widths of the hidden layers in $\phi_{\mathrm{ReLU}}$ and $\rho_{\mathrm{ReLU}}$ are $W_1$ and $W_3$ respectively. For the formal definition of $\phi_{\mathrm{ReLU}}$ and $\rho_{\mathrm{ReLU}}$, please refer to Appendix B. Then the function class with $\rho_{\mathrm{ReLU}}$ and $\phi_{\mathrm{ReLU}}$ as width-constrained ReLU networks is defined as

$$\mathcal{N}(W) = \left\{ f : \mathbb{R}^{N \times d} \to \mathbb{R} \,\middle|\, f(X) = \rho_{\mathrm{ReLU}}\left(\sum_{i=1}^{N} \phi_{\mathrm{ReLU}}(x_i)\right) \text{ with } \max_{i \in [3]} W_i \le W \right\}.$$

We would like to use functions in $\mathcal{N}(W)$ to approximate the self-attention function class

$$\mathcal{F} = \big\{ f : \mathbb{R}^{N\times d} \to \mathbb{R} \;\big|\; f(X) = \mathbb{I}_N^\top \mathrm{Att}(X, X, X) w \text{ for some } w \in [0,1]^d \big\}.$$

Figure 1(a) shows that $\rho_{\mathrm{ReLU}}(\sum_{i=1}^N \phi_{\mathrm{ReLU}}(x_i))$ first processes each channel with $\phi_{\mathrm{ReLU}}$, and the relationship between channels is only reasoned with $\rho_{\mathrm{ReLU}}$. The captured relationship in $\rho_{\mathrm{ReLU}}(\sum_{i=1}^N \phi_{\mathrm{ReLU}}(x_i))$ cannot be complex due to the simple structure of $\rho_{\mathrm{ReLU}}$. In contrast, the self-attention structure shown in Figure 1(b) first captures the relationship between channels with the self-attention structure and then weighs the results to derive the final output. Consequently, it is difficult to approximate the self-attention structure with $\rho_{\mathrm{ReLU}}(\sum_{i=1}^N \phi_{\mathrm{ReLU}}(x_i))$ due to its poor relational reasoning ability. This observation is formally quantified in the following theorem.

**Theorem 1.** *Let $W^*(\xi, d, \mathcal{F})$ be the smallest width of the neural network such that*

$$\forall\, f \in \mathcal{F},\ \exists\, g \in \mathcal{N}(W) \quad \textit{s.t.} \qquad \sup_{X\in[0,1]^{N\times d}} \big| f(X) - g(X) \big| \le \xi.$$

*With sufficient number of channels $N$, it holds that $W^*(\xi, d, \mathcal{F}) = \Omega(\exp(cd)\xi^{-1/4})$ for some $c > 0$.*

Theorem 1 shows that the fully-connected neural network cannot approximate the relational reasoning process in the self-attention mechanism unless the width is exponential in the input dimension. This exponential lower bound of the width of the fully-connected neural network implies that the relational reasoning process embedded within the self-attention structure is complicated, and it further motivates us to explicitly incorporate the self-attention structure in the neural networks in order to reason the complex relationship among the channels. We implement this in the following sections.

# 4 Offline Multi-Agent Reinforcement Learning with Set Transformers

## 4.1 Homogeneous Markov Decision Process

We consider the *homogeneous* MDP, where the transition kernel and the reward function are invariant to permutations of the agents, i.e., for any row-wise permutation function $\psi(\cdot)$, we have

$$P^*(\bar{S}' \,|\, \bar{S}, \bar{A}) = P^*\big(\psi(\bar{S}') \,\big|\, \psi(\bar{S}), \psi(\bar{A})\big) \quad \text{and} \quad r(\bar{S}, \bar{A}) = r\big(\psi(\bar{S}), \psi(\bar{A})\big)$$

for all $\bar{S}, \bar{S}' \in \mathcal{S}^N$ and $\bar{A} \in \mathcal{A}^N$. A key property of the homogeneous MDP is that there exists a permutation invariant optimal policy, and the corresponding state-value function and the action-value function are also permutation invariant [22].

**Proposition 1.** *For the cooperative homogeneous MDP, there exists an optimal policy that is permutation invariant. Also, for any permutation invariant policy $\pi$, the corresponding value function $V_{P^*}^\pi$ and action-value function $Q_{P^*}^\pi$ are permutation invariant.*

Thus, we restrict our attention to the class of permutation invariant policies $\Pi$, where $\pi(\bar{A} \,|\, \bar{S}) = \pi(\psi(\bar{A}) \,|\, \psi(\bar{S}))$ for all $\bar{A} \in \bar{\mathcal{A}}$, $\bar{S} \in \bar{\mathcal{S}}$, $\pi \in \Pi$ and all permutations $\psi$. For example, if $\pi(\bar{A} \,|\, \bar{S}) = \prod_{i=1}^N \mu(a_i \,|\, s_i)$ for some $\mu$, then $\pi$ is permutation invariant. An optimal policy is any $\pi^* \in \mathrm{argmax}_{\pi\in\Pi} V_{P^*}^\pi(\bar{S}_0)$.

## 4.2 Pessimistic Model-Free Offline Reinforcement Learning

In this subsection, we present a model-free algorithm, in which we adopt the transformer to estimate the action-value function. We also learn a policy based on such an estimate.

### 4.2.1 Function Approximation

In MARL, the relational reasoning is crucial in capturing the interaction between the different agents in the underlying MDP. Implementing the model-free algorithm on a homogeneous MDP, we want to estimate the action-value function, which should take the interaction among agents into account and reflect the homogeneity of the MDP in Proposition 1. Thus, the desired function class should satisfy these two conditions: (i) It is rich enough to capture the complex interaction between the agents. (ii) It is invariant to the permutation of the agents.

Motivated by the exponential width lower bound in Theorem 1, we explicitly embed the self-attention mechanism in the neural network structure. We take $X = [\bar{S}, \bar{A}] \in \mathbb{R}^{N \times d}$ as the input of the neural network, where $d = d_{\mathcal{S}} + d_{\mathcal{A}}$ and each agent corresponds to a channel in $X$. In the $i^{\text{th}}$ layer, as shown in Figure 4.2.1, we combine the self-attention mechanism $\mathrm{Att}(XW_{QK}^{(i)}, X, XW_V^{(i)})$ with the row-wise FeedForward (rFF) single-hidden layer neural network $\mathrm{rFF}(X, a^{(i)}, b^{(i)})$ with width $m$. We combine $W_Q^{(i)}$ and $W_K^{(i)}$ to $W_{QK}^{(i)}$ for ease of calculation, and $b^{(i)}$ and $a^{(i)}$ are the parameters of the first and second layer of rFF. The output of each layer is normalized by the row-wise normalization function $\Pi_{\mathrm{norm}}(\cdot)$, which projects each row of the input into the unit $\ell_p$-ball (for some $p \geq 1$). For the last layer, we derive the scalar estimate of the action-value function by averaging the outputs of all the channels, and the "clipping" function $\Pi_{V_{\max}}(x)$ is applied to normalize the output to $[-V_{\max}, V_{\max}]$. We note that such structures are also known as *set transformers* in [33]. For the formal definition of the transformer, please refer to Appendix C.

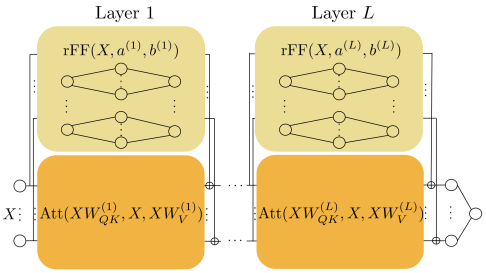



Figure 2: Structure of the transformer function class, where the row-wise feedforward function is specified as fully-connected networks.

The permutation invariance of the structure specified above can be easily shown with the permutation equivariance of the self-attention mechanism. We note that our results can be generalized to the multi-head attention structure, and the extensions are provided in Appendix N. We consider the transformer with the bounded parameters. For a pair of conjugate numbers $p, q \in \mathbb{R}$, i.e., $1/p + 1/q = 1$ and $p, q \geq 1$, the transformer function class with bounded parameters is defined as

$$\mathcal{F}_{\mathrm{tf}}(B) = \Big\{ g_{\mathrm{tf}}(X; W_{QK}^{1:L}, W_V^{1:L}, a^{1:L}, b^{1:L}, w) \,\Big|\, \big|a_{kj}^{(i)}\big| < B_a, \big\|b_{kj}^{(i)}\big\|_q < B_b, \\ \big\|W_{QK}^{(i)\top}\big\|_{p,q} < B_{QK}, \big\|W_V^{(i)\top}\big\|_{p,q} < B_V, \|w\|_q < B_w \text{ for } i \in [L], j \in [m], k \in [d] \Big\},$$

where $B = [B_a, B_b, B_{QK}, B_V, B_w]$ are the parameters of the function class, and $W_{QK}^{1:L}, W_V^{1:L}, a^{1:L}$ and $b^{1:L}$ are the stacked parameters in each layer. We only consider the non-trivial case where $B_a, B_b, B_{QK}, B_V, B_w$ are larger than one, otherwise the norms of the outputs decrease exponentially with growing depth. For ease of notation, we denote $\mathcal{F}_{\mathrm{tf}}(B)$ as $\mathcal{F}_{\mathrm{tf}}$ when the parameters are clear.

### 4.2.2 Algorithm

We modify the single-agent offline RL algorithm in [19] to accommodate the multi-agent case with the transformer approximators, but the analysis is quite different from that in [19]. Given the dataset $\mathcal{D} = \{(\bar{S}_i, \bar{A}_i, r_i, \bar{S}_i')\}_{i=1}^n$, we define the mismatch between two functions $f$ and $\tilde{f}$ on $\mathcal{D}$ for a fixed policy $\pi$ as $\mathcal{L}(f, \tilde{f}, \pi; \mathcal{D}) = \frac{1}{n} \sum_{(\bar{S}, \bar{A}, \bar{r}, \bar{S}') \in \mathcal{D}} (f(\bar{S}, \bar{A}) - \bar{r} - \gamma \tilde{f}(\bar{S}', \pi))^2$, then the Bellman error of a function $f$ with respect to the policy $\pi$ is defined as $\mathcal{E}(f, \pi; \mathcal{D}) = \mathcal{L}(f, f, \pi; \mathcal{D}) - \inf_{\tilde{f} \in \mathcal{F}_{\mathrm{tf}}} \mathcal{L}(\tilde{f}, f, \pi; \mathcal{D})$.

For a fixed policy $\pi$, we construct the confidence region of the action-value function of $\pi$ by selecting the functions in $\mathcal{F}_{\mathrm{tf}}$ with the $\varepsilon$-controlled Bellman error. We regard the function attaining the minimum in the confidence region as the estimate of the action-value function of the policy; this reflects the terminology "pessimism". Then the optimal policy is learned by maximizing the action-value function estimate. The algorithm can be written formally as

$$\hat{\pi} = \operatorname*{argmax}_{\pi \in \Pi} \min_{f \in \mathcal{F}(\pi, \varepsilon)} f(\bar{S}_0, \pi), \quad \text{where} \quad \mathcal{F}(\pi, \varepsilon) = \big\{ f \in \mathcal{F}_{\mathrm{tf}}(B) \,\big|\, \mathcal{E}(f, \pi; \mathcal{D}) \leq \varepsilon \big\}. \tag{1}$$

The motivation for the pessimism originates from the *distribution shift*, where the induced distribution of the learned policy is different from the sampling distribution $\nu$. Such an issue is severe when there is no guarantee that the sampling distribution $\nu$ supports the visitation distribution $d_{P^*}^{\pi^*}$ induced by the optimal policy $\pi^*$. In fact, the algorithm in Eqn. (1) does not require the *global* coverage of the sampling distribution $\nu$, where the global coverage means that $d_{P^*}^{\pi}(\bar{S}, \bar{A})/\nu(\bar{S}, \bar{A})$ is upper bounded by some constant for all $(\bar{S}, \bar{A}) \in \bar{\mathcal{S}} \times \bar{\mathcal{A}}$ and all $\pi \in \Pi$. Instead, it only requires *partial* coverage, and the mismatch between the distribution induced by the optimal policy $d_{P^*}^{\pi^*}$ and the sampling distribution $\nu$ is captured by

$$C_{\mathcal{F}_{\mathrm{tf}}} = \max_{f \in \mathcal{F}_{\mathrm{tf}}} \mathbb{E}_{d_{P^*}^{\pi^*}} \big[ \big( f(\bar{S}, \bar{A}) - \mathcal{T}^{\pi^*} f(\bar{S}, \bar{A}) \big)^2 \big] / \mathbb{E}_{\nu} \big[ \big( f(\bar{S}, \bar{A}) - \mathcal{T}^{\pi^*} f(\bar{S}, \bar{A}) \big)^2 \big]. \tag{2}$$

We note that $C_{\mathcal{F}_{\rm tf}} \leq \max_{(\bar{S},\bar{A})\in\bar{\mathcal{S}}\times\bar{\mathcal{A}}} d_{P^*}^{\pi^*}(\bar{S},\bar{A})/\nu(\bar{S},\bar{A})$, so the suboptimality bound involving $C_{\mathcal{F}_{\rm tf}}$ in Theorem 2 is tighter than the bound requiring global convergence [35]. Similar coefficients also appear in many existing works such as [19] and [36].

#### 4.2.3 Generalization Error Bound

In the analysis of the algorithm in Eqn. (1), we first derive a generalization error bound of the estimate of the Bellman error using the PAC-Bayesian framework [37, 38].

**Proposition 2.** *Let $\bar{B} = B_V B_{QK} B_a B_b B_w$. For all $f, \tilde{f} \in \mathcal{F}_{\rm tf}(B)$ and all policies $\pi \in \Pi$, with probability at least $1-\delta$, we have*

$$\begin{aligned}&\left|\mathbb{E}_\nu\Big[\big(f(\bar{S},\bar{A}) - \mathcal{T}^\pi \tilde{f}(\bar{S},\bar{A})\big)^2\Big] - \mathcal{L}(f,\tilde{f},\pi;\mathcal{D}) + \mathcal{L}(\mathcal{T}^\pi\tilde{f},\tilde{f},\pi;\mathcal{D})\right| \\ &\leq \frac{1}{2}\mathbb{E}_\nu\Big[\big(f(\bar{S},\bar{A}) - \mathcal{T}^\pi \tilde{f}(\bar{S},\bar{A})\big)^2\Big] + O\left(\frac{V_{\max}^2}{n}\left[mL^2d^2\log\frac{mdL\bar{B}n}{V_{\max}} + \log\frac{\mathcal{N}(\Pi,1/n,d_\infty)}{\delta}\right]\right).\end{aligned}$$

For ease of notation, we define $e(\mathcal{F}_{\rm tf},\Pi,\delta,n)$ to be $n$ times the second term of the generalization error bound. We compare our generalization error bound in Proposition 2 with [9, Theorem 4.6]. For the dependence on the number of agents $N$, the result in [9, Theorem 4.6] shows that the logarithm of the covering number of the transformer function class is logarithmic in $N$. Combined with the use of the Dudley integral [39], [9, Theorem 4.6] implies that the generalization error bound is logarithmic in $N$. In contrast, our result is independent of $N$. This superiority is attributed to our use of the PAC-Bayesian framework, in which we measure the distance between different functions using the KL divergence of the distributions on the function parameter space. For the transformer structure, the size of the parameter space is independent of the number of agents $N$, which helps us to remove the dependence on $N$.

For the dependence on the depth $L$ of the neural network, [9, Theorem 4.6] shows that the logarithm of the covering number of the transformer function class scales exponentially in $L$. In contrast, Proposition 2 shows that the generalization bound is *polynomial* in $L$. We note that Proposition 2 does not contradict the exponential dependence shown in [40, 41], since we implement the layer normalization to restrict the range of the output. As a byproduct, Proposition 2 shows that the invariant of the layer normalization adopted in our paper can greatly reduce the dependence of the generalization error on the depth of the neural network $L$.

#### 4.2.4 Bound on the Suboptimality Gap

Before stating the suboptimality bound, We require two assumptions on $\mathcal{F}_{\rm tf}$ and the sampling distribution $\nu$. We first state the standard regularity assumption of the transformer function class.

**Assumption 1.** *For any $\pi \in \Pi$, we have $\inf_{f\in\mathcal{F}_{\rm tf}} \sup_{\mu\in d_\Pi} \mathbb{E}_\mu[(f(\bar{S},\bar{A}) - \mathcal{T}^\pi f(\bar{S},\bar{A}))^2] \leq \varepsilon_{\mathcal{F}}$ and $\sup_{f\in\mathcal{F}_{\rm tf}} \inf_{\tilde{f}\in\mathcal{F}_{\rm tf}} \mathbb{E}_\nu[(\tilde{f}(\bar{S},\bar{A}) - \mathcal{T}^\pi f(\bar{S},\bar{A}))^2] \leq \varepsilon_{\mathcal{F},\mathcal{F}}$, where $d_\Pi = \{\mu \mid \exists\, \pi \in \Pi$ s.t. $\mu = d_{P^*}^\pi\}$ is the set of distributions of the state and the action pair induced by any policy $\pi \in \Pi$.*

This assumption, including the *realizability* and the *completeness*, states that for any policy $\pi \in \Pi$ there is a function in the transformer function class $\mathcal{F}_{\rm tf}$ such that the Bellman error is controlled by $\varepsilon_{\mathcal{F}}$, and the transformer function class is approximately closed under the Bellman operator $\mathcal{T}^\pi$ for any $\pi \in \Pi$. In addition, we require that the mismatch between the sampling distribution and the visitation distribution of the optimal policy is bounded.

**Assumption 2.** *For the sampling distribution $\nu$, the coefficient $C_{\mathcal{F}_{\rm tf}}$ defined in Eqn.* (2) *is finite.*

We note that similar assumptions also appear in many existing works [19, 36]. Then the suboptimality gap of the learned policy $\hat{\pi}$ can be upper bounded as the following.

**Theorem 2.** *If Assumptions 1 and 2 hold, and we take $\varepsilon = 3\varepsilon_{\mathcal{F}}/2 + 2e(\mathcal{F}_{\rm tf},\Pi,\delta,n)/n$, then with probability at least $1-\delta$, the suboptimality gap of the policy derived in the algorithm shown in Eqn.* (1) *is upper bounded as*

$$V_{P^*}^{\pi^*}(\bar{S}_0) - V_{P^*}^{\hat{\pi}}(\bar{S}_0) \leq O\left(\frac{\sqrt{C_{\mathcal{F}_{\rm tf}}}\tilde{\varepsilon}}{1-\gamma} + \frac{V_{\max}\sqrt{C_{\mathcal{F}_{\rm tf}}}}{(1-\gamma)\sqrt{n}}\sqrt{mL^2d^2\log\frac{mdL\bar{B}n}{V_{\max}} + \log\frac{2\mathcal{N}(\Pi,1/n,d_\infty)}{\delta}}\right),$$

*where $d = d_{\mathcal{S}} + d_{\mathcal{A}}$, $\tilde{\varepsilon} = \varepsilon_{\mathcal{F}} + \varepsilon_{\mathcal{F},\mathcal{F}}$, and $\bar{B}$ is defined in Proposition 2.*

Theorem 2 shows that the upper bound of the suboptimality gap does not scale with the number of agents $N$, which demonstrates that the proposed model-free algorithm breaks the curse of many agents. We note that the model-free offline/batch MARL with homogeneous agents has been studied in [8] and [22], and the suboptimality upper bounds in [8, Theorem 1] and [22, Theorem 4.1] are also independent of $N$. However, these works adopt the mean-field approximation of the original MDP, in which the influence of all the other agents on a specific agent is only coarsely considered through the distribution of the state. The approximation error between the action-value function of the mean-field MDP and that of the original MDP is not analyzed therein. Thus, the independence of $N$ in their works comes with the cost of the poor relational reasoning ability and the unspecified approximation error. In contrast, we analyze the suboptimality gap of the learned policy in the original MDP, and the interaction among agents is captured by the transformer network.

### 4.3 Pessimistic Model-based Offline Reinforcement Learning

In this subsection, we present the model-based algorithm, where we adopt the transformer to estimate the system dynamics and learn the policy based on such an estimate.

#### 4.3.1 Neural Nonlinear Regulator

In this section, we consider the Neural Nonlinear Regulator (NNR), in which we use the transformer to estimate the system dynamics. The ground truth transition $P^*(\bar{S}' \mid \bar{S}, \bar{A})$ is defined as $\bar{S}' = F^*(\bar{S}, \bar{A}) + \bar{\varepsilon}$, where $F^*$ is a nonlinear function, $\bar{\varepsilon} = [\varepsilon_1, \ldots, \varepsilon_N]^\top$ is the noise, and $\varepsilon_i \sim \mathcal{N}(0, \sigma^2 I_{d\times d})$ for $i \in [N]$ are independent random vectors. We note that the function $F^*$ and the transition kernel $P^*$ are equivalent, and we denote the transition kernel corresponding to the function $F$ as $P_F$. Since the transition kernel $P^*(\bar{S}' \mid \bar{S}, \bar{A})$ is permutation invariant, $F^*$ should be permutation equivariant, i.e., $F^*(\psi(\bar{S}), \psi(\bar{A})) = \psi(F^*(\bar{S}, \bar{A}))$ for all row-wise permutation functions $\psi(\cdot)$.

We take $X = [\bar{S}, \bar{A}] \in \mathbb{R}^{N\times d}$ as the input of the network and adopt a similar network structure as the transformer specified in Section 4.2.1. However, to predict the next state instead of the action-value function with the transformer, we remove the average aggregation module in the final layer of the structure in Section 4.2.1. Please refer to Appendix C for the formal definition. The permutation equivariance of the proposed transformer structure can be easily proved with the permutation equivariance of the self-attention mechanism. We consider the transformer function class with bounded parameters, which is defined as

$$\mathcal{M}_{\mathrm{tf}}(B') = \Big\{ F_{\mathrm{tf}}(X; W_{QK}^{1:L}, W_V^{1:L}, a^{1:L}, b^{1:L}) \,\Big|\, \big|a_{kj}^{(i)}\big| < B_a, \big\|b_{kj}^{(i)}\big\|_2 < B_b, \\ \big\|W_{QK}^{(i)\top}\big\|_{\mathrm{F}} < B_{QK}, \big\|W_V^{(i)\top}\big\|_{\mathrm{F}} < B_V \text{ for } i \in [L], j \in [m], k \in [d] \Big\},$$

where $B' = [B_a, B_b, B_{QK}, B_V]$ is the vector of parameters of the function class. We denote $\mathcal{M}_{\mathrm{tf}}(B')$ as $\mathcal{M}_{\mathrm{tf}}$ when the parameters are clear from the context.

#### 4.3.2 Algorithm

Given the offline dataset $\mathcal{D} = \{(\bar{S}_i, \bar{A}_i, r_i, \bar{S}_i')\}_{i=1}^n$, we first derive the MLE of the system dynamics. Next, we learn the optimal policy according to the confidence region of the dynamics that are constructed around the MLE. The term "pessimism" is reflected in the procedure that we choose the system dynamics that induce the *smallest* value function, i.e.,

$$\hat{F}_{\mathrm{MLE}} = \operatorname*{argmin}_{F\in\mathcal{M}_{\mathrm{tf}}} \frac{1}{n}\sum_{i=1}^n \big\|\bar{S}_i' - F(\bar{S}_i, \bar{A}_i)\big\|_{\mathrm{F}}^2 \quad \text{and} \quad \hat{\pi} = \operatorname*{argmax}_{\pi\in\Pi} \min_{F\in\mathcal{M}_{\mathrm{MLE}}(\zeta)} V_{P_F}^{\pi}(\bar{S}_0), \tag{3}$$

where $\mathcal{M}_{\mathrm{MLE}}(\zeta) = \{F \in \mathcal{M}_{\mathrm{tf}}(B') \mid 1/n \cdot \sum_{i=1}^n \mathrm{TV}(P_F(\cdot \mid \bar{S}_i, \bar{A}_i), \hat{P}_{\mathrm{MLE}}(\cdot \mid \bar{S}_i, \bar{A}_i))^2 \le \zeta\}$ is the confidence region, which has a closed-form expression in terms of the difference between $F$ and $\hat{F}_{\mathrm{MLE}}$ as stated in in Appendix D. The transition kernel induced by $\hat{F}_{\mathrm{MLE}}$ is denoted as $\hat{P}_{\mathrm{MLE}}$. The parameter $\zeta$ is used to measure the tolerance of estimation error of the system dynamics, and it is set to according to the parameters of the function class such that $F^*$ belongs to $\mathcal{M}_{\mathrm{MLE}}(\zeta)$ with high probability.

Similar to the model-free algorithm, the model-based algorithm specified in Eqn. (3) does not require global coverage. Instead, the mismatch between the distribution induced by the optimal policy $d_{P^*}^{\pi^*}$ and the sampling distribution $\nu$ is captured by the constant

$$C_{\mathcal{M}_{\mathrm{tf}}} = \max_{F \in \mathcal{M}_{\mathrm{tf}}} \mathbb{E}_{d_{P^*}^{\pi^*}} \left[ \mathrm{TV}\big(P_F(\cdot \mid \bar{S}, \bar{A}), P^*(\cdot \mid \bar{S}, \bar{A})\big)^2 \right] \Big/ \mathbb{E}_{\nu} \left[ \mathrm{TV}\big(P_F(\cdot \mid \bar{S}, \bar{A}), P^*(\cdot \mid \bar{S}, \bar{A})\big)^2 \right]. \quad (4)$$

We note that $C_{\mathcal{M}_{\mathrm{tf}}} \leq \max_{(\bar{S},\bar{A}) \in \bar{\mathcal{S}} \times \bar{\mathcal{A}}} d_{P^*}^{\pi^*}(\bar{S}, \bar{A}) / \nu(\bar{S}, \bar{A})$, so the suboptimality bound involving $C_{\mathcal{P}_{\mathcal{F}_{\mathrm{tf}}}}$ in Theorem 3 is tighter than the bound requiring global convergence. Similar coefficients also appear in many existing works such as [42] and [20].

#### 4.3.3 Analysis of the Maximum Likelihood Estimate

Every $F \in \mathcal{M}_{\mathrm{MLE}}(\zeta)$ is near to the MLE in the total variation sense and thus well approximates the ground truth system dynamics. Therefore, to derive an upper bound of the suboptimality gap of the learned policy, we first analyze the convergence rate of the MLE $\hat{P}_{\mathrm{MLE}}$ to $P^*$.

**Proposition 3.** *Let $\tilde{B} = B_V B_{QK} B_a B_b$. For the maximum likelihood estimate $\hat{P}_{\mathrm{MLE}}$ in Eqn.* (3)*, the following inequality holds with probability at least $1 - \delta$,*

$$\mathbb{E}_{\nu} \Big[ \mathrm{TV}\big(P^*(\cdot \mid \bar{S}, \bar{A}), \hat{P}_{\mathrm{MLE}}(\cdot \mid \bar{S}, \bar{A})\big)^2 \Big] \leq O\bigg( \frac{1}{n} m L^2 d^2 \log\big(NLmd\tilde{B}n\big) + \frac{1}{n} \log \frac{1}{\delta} \bigg).$$

We define $e'(\mathcal{M}_{\mathrm{tf}}, n)$ to be $n$ times the total variation bound. Proposition 3 shows that the total variation estimation error is polynomial in the depth of the neural network $L$. However, different from the model-free RL results in Section 4.2, the estimation error of MLE $\hat{P}_{\mathrm{MLE}}$ is logarithmic in the number of agents $N$. We note that this logarithm dependency on $N$ comes from the fact that $\mathrm{TV}(P^*(\cdot \mid \bar{S}, \bar{A}), \hat{P}_{\mathrm{MLE}}(\cdot \mid \bar{S}, \bar{A}))$ measures the distance between two transition kernels that involves the states of $N$ agents, different from the scalar estimate of the value function in Section 4.2. To prove the result, we adopt a PAC-Bayesian framework to analyze the convergence rate of MLE, which is inspired by the analysis of density estimation [43]; more details are presented in Appendix J.

#### 4.3.4 Bound on the Suboptimality Gap

To analyze the error of the learned model, we make the following realizability assumption.

**Assumption 3.** *The nominal system dynamics belongs to the function class $\mathcal{M}_{\mathrm{tf}}$, i.e., $F^* \in \mathcal{M}_{\mathrm{tf}}(B')$.*

In addition, we require that the mismatch between the sampling distribution and the visitation distribution of the optimal policy is bounded.

**Assumption 4.** *For the sampling distribution $\nu$, the coefficient $C_{\mathcal{M}_{\mathrm{tf}}}$ defined in* (4) *is finite.*

We note that these two assumptions are also made in many existing works, e.g., [20, 21].

**Theorem 3.** *If Assumptions 3 and 4 hold, and we take $\zeta = c_1 e'(\mathcal{M}_{\mathrm{tf}}, n)/n$ for some constant $c_1 > 0$, then with probability at least $1 - \delta$, the suboptimality gap of the policy learned in the algorithm in Eqn.* (3) *is upper bounded as*

$$V_{P^*}^{\pi^*}(\bar{S}_0) - V_{P^*}^{\hat{\pi}}(\bar{S}_0) \leq O\bigg( \frac{V_{\max}}{(1-\gamma)^2} \sqrt{C_{\mathcal{M}_{\mathrm{tf}}} \bigg( \frac{1}{n} m L^2 d^2 \log\big(NLmd\tilde{B}n\big) + \frac{1}{n} \log \frac{1}{\delta} \bigg)} \bigg),$$

*where $d = d_{\mathcal{S}} + d_{\mathcal{A}}$, and $\tilde{B}$ is defined in Proposition 3.*

Theorem 3 presents an upper bound on the suboptimality gap of the offline model-based RL with the transformer approximators. The suboptimality gap depends on the number of agents only as $O(\sqrt{\log N})$, which shows that the proposed model-based MARL algorithm mitigates the curse of many agents. This weak dependence on $N$ originates from measuring the distance between two system dynamics of $N$ agents in the learning of the dynamics. To the best of our knowledge, there is no prior work on analyzing the model-based algorithm for the homogeneous MARL, even from the mean-field approximation perspective. The proof of Theorem 3 leverages novel analysis of the MLE in Proposition 3. For more details, please refer to Appendix H. Concluding remarks are provided in Appendix A.

# References


[1] D. Wang, R. Walters, X. Zhu, and R. Platt. Equivariant Q Learning in Spatial Action Spaces. In *Conference on Robot Learning*, pages 1713–1723. PMLR, 2022.

[2] D. Wang, R. Walters, and R. Platt. SO(2) -equivariant reinforcement learning. *arXiv preprint arXiv:2203.04439*, 2022.

[3] Y. Tang and D. Ha. The sensory neuron as a transformer: Permutation-invariant neural networks for reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[5] K. Xu, Y. Zhang, D. Ye, P. Zhao, and M. Tan. Relation-aware transformer for portfolio policy learning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4647–4653, 2021.

[6] L. Wang, Z. Yang, and Z. Wang. Breaking the curse of many agents: Provable mean embedding q-iteration for mean-field reinforcement learning. In *International Conference on Machine Learning*, pages 10092–10103. PMLR, 2020.

[7] K. Menda, Y. Chen, J. Grana, J. W. Bono, B. D. Tracey, M. J. Kochenderfer, and D. Wolpert. Deep reinforcement learning for event-driven multi-agent decision processes. *IEEE Transactions on Intelligent Transportation Systems*, 20(4):1259–1268, 2018.

[8] M. Chen, Y. Li, E. Wang, Z. Yang, Z. Wang, and T. Zhao. Pessimism meets invariance: Provably efficient offline mean-field multi-agent RL. *Advances in Neural Information Processing Systems*, 34, 2021.

[9] B. Edelman, S. Goel, S. Kakade, and C. Zhang. Inductive biases and variable creation in self-attention mechanisms. *arXiv preprint arXiv:2110.10090*, 2021.

[10] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. Deep sets. *Advances in Neural Information Processing Systems*, 30, 2017.

[11] J. Chen and N. Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pages 1042–1051. PMLR, 2019.

[12] A. Antos, C. Szepesvári, and R. Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129, 2008.

[13] O. Nachum, B. Dai, I. Kostrikov, Y. Chow, L. Li, and D. Schuurmans. Algaedice: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*, 2019.

[14] Y. Duan, C. Jin, and Z. Li. Risk bounds and rademacher complexity in batch reinforcement learning. In *International Conference on Machine Learning*, pages 2892–2902. PMLR, 2021.

[15] T. Xie and N. Jiang. Q* approximation schemes for batch reinforcement learning: A theoretical comparison. In *Conference on Uncertainty in Artificial Intelligence*, pages 550–559. PMLR, 2020.

[16] S. Ross and J. A. Bagnell. Agnostic system identification for model-based reinforcement learning. *arXiv preprint arXiv:1203.1007*, 2012.

[17] Y. Jin, Z. Yang, and Z. Wang. Is pessimism provably efficient for offline RL? In *International Conference on Machine Learning*, pages 5084–5096. PMLR, 2021.

[18] N. Rajaraman, L. Yang, J. Jiao, and K. Ramchandran. Toward the fundamental limits of imitation learning. *Advances in Neural Information Processing Systems*, 33:2914–2924, 2020.

[19] T. Xie, C. Cheng, N. Jiang, P. Mineiro, and A. Agarwal. Bellman-consistent pessimism for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[20] J. Chang, M. Uehara, D. Sreenivas, R. Kidambi, and W. Sun. Mitigating covariate shift in imitation learning via offline data with partial coverage. *Advances in Neural Information Processing Systems*, 34, 2021.

[21] M. Uehara and W. Sun. Pessimistic model-based offline reinforcement learning under partial coverage. *arXiv preprint arXiv:2107.06226*, 2021.

[22] Y. Li, L. Wang, J. Yang, E. Wang, Z. Wang, T. Zhao, and H. Zha. Permutation invariant policy optimization for mean-field multi-agent reinforcement learning: A principled approach. *arXiv preprint arXiv:2105.08268*, 2021.

[23] J. Sokolic, R. Giryes, G. Sapiro, and M. Rodrigues. Generalization error of invariant classifiers. In *Artificial Intelligence and Statistics*, pages 1094–1103. PMLR, 2017.

[24] A. Sannai, M. Imaizumi, and M. Kawano. Improved generalization bounds of group invariant/equivariant deep networks via quotient feature spaces. In *Uncertainty in Artificial Intelligence*, pages 771–780. PMLR, 2021.

[25] D. Jakubovitz, R. Giryes, and M. R. D. Rodrigues. Generalization error in deep learning. In *Compressed sensing and its applications*, pages 153–193. Springer, 2019.

[26] S. Zhu, B. An, and F. Huang. Understanding the generalization benefit of model invariance from a data perspective. *Advances in Neural Information Processing Systems*, 34, 2021.

[27] Anonymous. An analysis of attention via the lens of invariance: Approximation, generalization, and optimization. *Technical Report*, 2022.

[28] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[29] M. M. Naseer, K. Ranasinghe, S. H Khan, M. Hayat, F. Shahbaz Khan, and M. Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34, 2021.

[30] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *In International Conference on Learning Representations*, 2021.

[32] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. H. Tay, J. Feng, and S. Yan. Tokens-to-token vit: Training vision transformers from scratch on Imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021.

[33] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pages 3744–3753. PMLR, 2019.

[34] S. Sonoda and N. Murata. Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*, 43(2):233–268, 2017.

[35] M. Uehara, J. Huang, and N. Jiang. Minimax weight and Q-function learning for off-policy evaluation. In *International Conference on Machine Learning*, pages 9659–9668. PMLR, 2020.

[36] M. Yin, Y. Duan, M. Wang, and Y.-X. Wang. Near-optimal offline reinforcement learning with linear representation: Leveraging variance information with pessimism. *arXiv preprint arXiv:2203.05804*, 2022.

[37] D. A. McAllester. Some PAC-Bayesian theorems. *Machine Learning*, 37(3):355–363, 1999.

[38] D. A. McAllester. Simplified PAC-Bayesian margin bounds. In *Learning Theory and Kernel Machines*, pages 203–215. Springer, 2003.

[39] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT press, 2018.

[40] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in Neural Information Processing Systems*, 30, 2017.

[41] B. Neyshabur, S. Bhojanapalli, and N. Srebro. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

[42] W. Sun, N. Jiang, A. Krishnamurthy, A. Agarwal, and J. Langford. Model-based RL in contextual decision processes: PAC bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*, pages 2898–2933. PMLR, 2019.

[43] T. Zhang. From $\varepsilon$-entropy to KL-entropy: Analysis of minimum information complexity density estimation. *The Annals of Statistics*, 34(5):2180–2210, 2006.

[44] L. Devroye, A. Mehrabian, and T. Reddad. The total variation distance between high-dimensional Gaussians. *arXiv preprint arXiv:1810.08693*, 2018.

[45] M. Telgarsky. Representation benefits of deep feedforward networks. *arXiv preprint arXiv:1509.08101*, 2015.

[46] A. Agarwal, S. Kakade, A. Krishnamurthy, and W. Sun. Flambe: Structural complexity and representation learning of low rank MDPs. *Advances in Neural Information Processing Systems*, 33:20095–20107, 2020.

[47] M. Tsagris, C. Beneki, and H. Hassani. On the folded normal distribution. *Mathematics*, 2(1):12–28, 2014.

[48] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, pages 531–540. PMLR, 2018.

[49] M. J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*, volume 48. Cambridge University Press, 2019.

## Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [N/A]
(b) Did you mention the license of the assets? [N/A]
(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]