Dependency Structure Search Bayesian Optimization for Decision Making Models

Anonymous authors
Paper under double-blind review

Abstract

Many approaches for optimizing decision making models rely on gradient based methods requiring informative feedback from the environment. However, in the case where such feedback is sparse or uninformative, such approaches may result in poor performance. Derivative-free approaches such as Bayesian Optimization mitigate the dependency on the quality of gradient feedback, but are known to scale poorly in the high-dimension setting of complex decision making models. This problem is exacerbated if the model requires interactions between several agents cooperating to accomplish a shared goal. To address the dimensionality challenge, we propose a compact multi-layered architecture modeling the dynamics of agent interactions through the concept of role. We introduce Dependency Structure Search Bayesian Optimization to efficiently optimize the multi-layered architecture parameterized by a large number of parameters, and give the first improved regret bound in additive high-dimensional Bayesian Optimization since Mutny & Krause (2018). Our approach shows strong empirical results under malformed or sparse reward.

1 Introduction

Decision Making Models choose sequences of actions to accomplish a goal. Multi-Agent Decision Making Models choose actions for multiple agents working together towards a shared goal. Multi-Agent Reinforcement Learning (MARL) has emerged as a competitive approach for optimizing Decision Making Models in the multi-agent setting.[1] MARL optimizes a *policy* under the partially observable Markov Decision Process (POMDP) framework, where decision making happens in an *environment* determined by a set of possible states and actions, and the *reward* for an action is conditioned upon the partially observable state of the environment. A policy forms a set of decision-making rules capturing the most rewarding actions in a given state. MARL utilizes gradient-based methods requiring a differentiable policy and informative gradients to make progress. This restriction requires the usage of large gradient-friendly policy representations (e.g., neural networks) and informative reward feedback from the environment (Pathak et al., 2017; Qian & Yu, 2021) which may not always be present. In addition, gradient-based methods are susceptible to falling into local maxima.

The confluence of computationally expensive policy representations, uninformative reward, and susceptibility to local maxima motivate this work. In the context of memory-constrained devices such as Internet of Things (IoT) devices (Merenda et al., 2020), utilizing large neural networks is infeasible. Secondly, in environments with sparse reward feedback, training these networks with RL presents significant challenges due to unhelpful policy gradients. Finally, the possibility of *globally optimizing* a compact policy for memory-constrained systems is appealing due to its strong performance guarantees.

We propose the usage of Bayesian Optimization (BO) for multi-agent policy search (MAPS) that makes progress on overcoming these issues in Decision Making Models. Since BO is a gradient-free optimizer capable of searching globally, applying BO to MAPS both ensures global searching of the policy, and overcomes poor gradient behavior in the reward function (Qian & Yu, 2021). The chief challenge in BO for MAPS is

[1] We include an overview of approaches in Decision Making Models in Section 3.

the high dimensionality of complex multi-agent interactions. However, our proposed setting of optimizing compact policies suitable for *memory-constrained* devices enables the possibility of overcoming this limitation.

A significant degree of high-dimensional multi-agent interactions exist in MAPS. For example, considering an autonomous drone delivery system, several agents (i.e., drones) must work together to maximize the throughput of deliveries. In doing so, these agents may separate themselves into different roles, for example, long-distance or short-distance deliveries. The optimal policy for each role may be significantly different due to distances to recharging base stations (e.g., drones must conserve battery). In forming the optimal policy, the *interaction* between agents must be considered to both optimally divide the task between the drones, as well as coordinate actions between drones (e.g., collision avoidance). These interactions may change over time. For example, a drone must avoid collision with nearby drones, which changes as it moves through the environment. With many agents, these interactions become more complex.

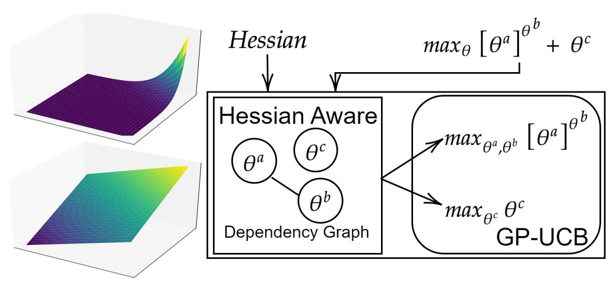

Figure 1: Left, above, plot of $f(x, y) = x^y$; below, plot of $f(x, y) = x + y$. The curvature of *additively constructed functions* is zero; *non-zero curvature* indicates dependency among input variables. Right, examining the Hessian learns the dependency structure which decomposes complex problems into simpler problems solved by GP-UCB.

To tackle the high-dimensional complexity, we utilize specific multi-agent abstractions of *role* and *role interaction*. In role-based multi-agent interactions, an agent's policy depends on its current role and sparse interactions with other agents. By simplifying the policy space with these abstractions, we increase its tractability for global optimization by BO and inherit the strong empirical performance demonstrated by these approaches. We realize this simplification of the policy space by expressing the role abstraction and role interaction abstractions as immutable portions of the policy space, which are not searched over during policy optimization. To achieve this, we use a higher-order model (HOM) which *generates* a policy model. The HOM is divided into immutable instructions (i.e., algorithms) corresponding to the abstractions of the role and role interaction and mutable parameters that are used to generate (GEN) a policy model during evaluation.

To optimize the HOM, we specialize BO by exploiting task-specific structures. A promising avenue of High-dimensional Bayesian Optimization (HDBO) is through additive decomposition. Additive decomposition separates a high-dimensional optimization problem into several independent low-dimensional sub-problems (Duvenaud et al., 2011; Kandasamy et al., 2015). These sub-problems are independently solved thus reducing the complexity of high dimensional optimization. However, a significant challenge in additive decomposition is *learning the independence structure* which is unknown a-priori. Learning the additive decomposition is accomplished using stochastic sampling such as Gibbs sampling (Kandasamy et al., 2015; Rolland et al., 2018; Han et al., 2020) which is known to have poor performance in high dimensions (Johnson et al., 2013; Barbos et al., 2017).

In our work, we overcome this shortcoming by observing the GEN process of the HOM. In particular, we can measure a surrogate Hessian during the GEN process which significantly simplifies the task of learning the additive structure. We term this approach Dependency Structure Search GP-UCB (DSS-GP-UCB) and visualize our approach in Fig. 1. Our proposed approach is also applicable to policy-search in the single-agent setting, showing its general-purpose applicability in Decision Making Models. In this work, we make the following contributions:

- We propose a parameter-efficient HOM for MAPS which is both expressive and compact. Our approach is made feasible by using specific abstractions of *roles* and *role interactions*.
- We propose DSS-GP-UCB, a variant of BO that simplifies the learning of dependency structure and provides strong regret guarantees which scale with $\mathcal{O}(\log(D))$ under reasonable assumptions.

Table 1: Summary of key notations.

Notation	Description
v	The objective function being optimized by Bayesian optimization
Θ	The domain for the objective function v
θ_t	A point in the domain Θ that is picked at time t
μ_T^k	The posterior mean (inferred after observations up to time $T-1$) at time T using the kernel k
$[\sigma_T^k]^2$	The posterior variance at time T using the kernel k
$r(\theta_t)$	The difference between the maxima of the function v in domain Θ, $v(\theta^*)$, and $v(\theta_t)$
R_T	The cumulative regret, $\sum_{t=1}^{T} r(\theta_t)$
Θ^a	Dimension a of the domain D
$\mathcal{G}_d$	A graph showing the dependencies between dimensions where edges exist between two dimensions if they are dependent
V_d	In the graph indicated by $\mathcal{G}_d$ the set of dimensions corresponding to Θ
E_d	In the graph indicated by $\mathcal{G}_d$ the set of edges corresponding to the dependencies between Θ
$\Theta^{(i)}$	Collection of dimensions indicated by (i) corresponding to a maximal clique in the graph $\mathcal{G}_d$
$k^{\Theta^{(i)}}$	A Gaussian process kernel correspond to the maximal clique (i)
k	The Gaussian process kernel for inference corresponding to the sum of $k^{\Theta^{(i)}}$: $k \triangleq \sum_i k^{\Theta^{(i)}}$
$v^{(i)}$	Under the additive assumption, it is assumed that $v = \sum_i v^{(i)}$ where each $v^{(i)}$ is sampled from $k^{\Theta^{(i)}}$
$\mathcal{U}(\Theta)$	A uniform random distribution over the domain Θ
$H(\theta_{t,h})$	A query to the Hessian at $\theta_{t,h}$
$\widetilde{\mathcal{G}_d}$	The graph corresponding to the detected dependency structure by querying the Hessian
Max-Cliques($\widetilde{\mathcal{G}_d}$)	A function computing the maximal cliques in the graph $\widetilde{\mathcal{G}_d}$
$\mathbf{s}$	The set of states of the cooperative multi-agent system where $\mathbf{s} \triangleq [\mathbf{s}^i]_{i=1,\ldots,n}$ and i denotes the index of the agent
$\mathbf{a}$	The set of actions taken by each agent where $\mathbf{a} \triangleq [\mathbf{a}^i]_{i=1,\ldots,n}$ and i denotes the index of the agent
$\mathbf{s}^{\alpha(i)}$	The state for agent a taking on the role $\alpha(i)$
$\mathbf{a}^{\alpha(i)}$	The action taken by agent a taking on the role $\alpha(i)$
$\Lambda^{\theta_{r,i}}$	An affinity function for taking on role i where r denotes it belonging to the part of the HOM for role assignment
$\Lambda^{\theta_{g,v}}$	An affinity function determining whether an edge exists during the interaction of roles in the HOM policy
$M^{\theta_{g,\eta}}$	The message passing function parameterized by $\theta_{g,\eta}$ for the role interaction message passing neural network
$U^{\theta_{g,e}}$	The action update function parameterized by $\theta_{g,e}$ for the role interaction message passing neural network

- We validate our approach on several multi-agent benchmarks and show our approach outperforms related works for compact models fit for memory-constrained scenarios. Our DSS-GP-UCB also overcomes poor gradient behavior in the reward function in multiple settings showing its effectiveness in Decision Making Models both in the single-agent and multi-agent settings.

2 Background

Bayesian Optimization: Bayesian optimization (BO) involves sequentially maximizing an unknown objective function $v : \Theta \to \mathbb{R}$. In each iteration $t = 1, \ldots, T$, an input query θ_t is evaluated to yield a noisy observation $y_t \triangleq v(\theta_t) + \epsilon$ with i. i. d. Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$. BO selects input queries to approach the global maximizer $\theta^* \triangleq \arg\max_{\theta \in \Theta} v(\theta)$ as rapidly as possible. This is achieved by minimizing *cumulative* regret $R_T \triangleq \sum_{t=1}^{T} r(\theta_t)$, where $r(\theta_t) \triangleq v(\theta^*) - v(\theta_t)$.

The belief of v is modeled by a *Gaussian process* (GP), denoted GP $(\mu(\theta), k(\theta, \theta'))$, that is, every finite subset of $\{v(\theta)\}_{\theta \in \Theta}$ follows a multivariate Gaussian distribution (Rasmussen & Williams, 2006). A GP is fully specified by its *prior* mean $\mu(\theta)$ and covariance $k(\theta, \theta')$ for all $\theta, \theta' \in \Theta$, which are, respectively, assumed w.l.o.g. to be $\mu(\theta) = 0$ and $k(\theta, \theta') \leq 1$. Given a vector $\mathbf{y}_T \triangleq [y_t]_{t=1,\ldots,T}^{\top}$ of noisy observations from evaluating v at input queries $\theta_1, \ldots, \theta_T \in \Theta$ after T iterations, the GP posterior belief of v at some input $\theta \in \Theta$ is a Gaussian with the following *posterior* mean $\mu_T^k(\theta)$ and variance $[\sigma_T^k]^2(\theta)$:

$$\mu_T^k(\theta) \triangleq \mathbf{k}_T^k(\theta)^{\top}(\mathbf{K}_T^k + \sigma^2\mathbf{I})^{-1}\mathbf{y}_T, \qquad \left[\sigma_T^k\right]^2(\theta) \triangleq k(\theta, \theta) - \mathbf{k}_T^k(\theta)^{\top}(\mathbf{K}_T^k + \sigma^2\mathbf{I})^{-1}\mathbf{k}_T^k(\theta) \tag{1}$$

where $\mathbf{K}_T^k \triangleq [k(\theta_t, \theta_{t'})]_{t,t'=1,\ldots,T}$ and $\mathbf{k}_T^k(\theta) \triangleq [k(\theta_t, \theta)]_{t=1,\ldots,T}^{\top}$. In each iteration t of BO, an input query $\theta_t \in \Theta$ is selected to maximize the GP-UCB acquisition function, $\theta_t \triangleq \arg\max_{\theta \in \Theta} \mu_{t-1}(\theta) + \sqrt{\beta_t}\sigma_{t-1}(\theta)$ (Srinivas et al., 2010) where β_t follows a well defined pattern.

Table 1 provides a summary of notations that are used frequently in paper.

3 Related work

Decision Making Models: Decision Making Models (Rizk et al., 2018; Roijers et al., 2013) determine actions taken by an agent or agents in order to achieve a goal. We focus on the POMDP setting and optimizing a policy to accumulate maximum reward while interacting with a partially observable environment (Shani et al., 2013). Many approaches exist which can be broadly categorized into direct policy search and reinforcement learning methods. Direct policy search (Heidrich-Meisner & Igel, 2008; Lizotte et al., 2007; Martinez-Cantin, 2017; Papavasileiou et al., 2021; Wierstra et al., 2008) searches the policy space in some efficient manner. Reinforcement learning (Arulkumaran et al., 2017; Fujimoto et al., 2018; Haarnoja et al., 2018; Lillicrap et al., 2015; Lowe et al., 2017; Mnih et al., 2015; Schulman et al., 2017) starts with a randomly initialized policy and *reinforces* rewarding behavior patterns to improve the policy.

Bayesian Optimization for Decision Making Models: BO has been utilized for direct policy search in the low dimensional setting (Lizotte et al., 2007; Wilson et al., 2014; Marco et al., 2016; Martinez-Cantin, 2017; von Rohr et al., 2018). However, these approaches have not scaled to the high dimensional setting. In more recent works, BO has been utilized to aid in local search methods similar to reinforcement learning (Akrour et al., 2017; Eriksson et al., 2019a; Wang et al., 2020a; Fröhlich et al., 2021; Müller et al., 2021). However, these approaches require evaluation of an inordinate number of policies typical of local search methods and do not provide regret guarantees. Recently, combinations of local and global search methods have been proposed (McLeod et al., 2018; Shekhar & Javidi, 2021). However, these approaches rely on informative and useful gradient information and have not been shown to scale to the high dimensional setting.

MARL for multi-agent decision making: A well-known approach for cooperative MARL is a combination of centralized training and decentralized execution (CTDE) (Oliehoek et al., 2008). The multi-agent interactions of CTDE methods can be implicitly captured by learning approximate models of other agents (Lowe et al., 2017; Foerster et al., 2018) or decomposing global rewards (Sunehag et al., 2017; Rashid et al., 2018; Son et al., 2019). However, these methods do not focus on how interactions are performed between agents. In MARL, the concept of *role* is often leveraged to enhance the flexibility of behavioral representation while controlling the complexity of the design of agents (Lhaksmana et al., 2018; Wang et al., 2020b; 2021b; Li et al., 2021). Our approach is related to the study of (Le et al., 2017a) where the interactions are also captured by role assignment. However, the approach operates on an imitation learning scenario, and the role assignment depends on the heuristic from domain knowledge. Another related field is Comm-MARL (Zhu et al., 2022; Shao et al., 2022; Liu et al., 2020; Peng et al., 2017; Das et al., 2019; Singh et al., 2019), where agents are allowed to communicate during policy execution to jointly decide on an action. In contrast, our approach utilizes both abstractions of role and role interaction in a HOM for a decision making model.

4 Design

We consider the problem of learning the joint policy of a set of n agents working cooperatively to solve a common task. Each agent i is associated with a state $\mathbf{s}^i \in \mathcal{S}^i$ with the global state represented as $\mathbf{s} \triangleq [\mathbf{s}^i]_{i=1,\dots,n}$. Each agent i cooperatively chooses an action $\mathbf{a}^i \in \mathcal{A}^i$ with the global action represented by $\mathbf{a} \triangleq [\mathbf{a}^i]_{i=1,\dots,n}$. Each state, action pair is associated with a *reward* function: $r(\mathbf{s}, \mathbf{a})$. In order to achieve the common task, a policy parameterized by θ: $\pi^\theta \triangleq \mathcal{S} \to \mathcal{A}$ governs the action taken by the agents, after observing state $\mathbf{s} \in \mathcal{S}$. The goal of RL is to learn the optimal policy parameters that maximizes the accumulation of rewards, $v(\theta)$, while acting in an unknown environment and receiving feedback through the resultant states and rewards.[2] We treat $v(\theta)$ as a black box function measuring the *value* of a policy and utilize BO to optimize θ.

[2]Further RL overview can be found in Arulkumaran et al. (2017).

4.1 Architectural design

To achieve a compact and tractable policy space, we consider policies under the useful abstractions of *role* and *role interaction*. These abstractions have consistently shown strong performance in multi-agent tasks. Therefore we can simplify the policy space by limiting it to only policies using these abstractions.

As role and role interaction are immutable abstractions within our policy space, we express them as *static algorithms* which are not searched over during policy optimization. These algorithms take as input parameters which are mutable and searched over during policy optimization. This combination of immutable instructions, and mutable parameters reduces the size of the search space,[3] yet is still able to express policies which conform to the role and role interaction abstractions.

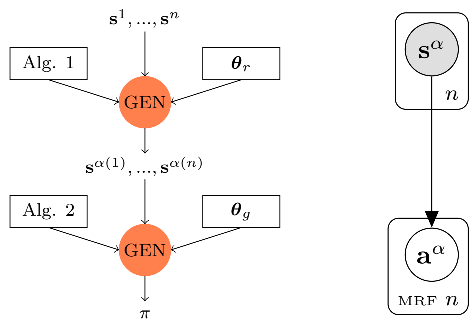

Figure 2: Left: HOM architecture. "Alg. 1/Alg. 2" are Algorithms 1 and 2 respectively. GEN uses θ_r and θ_g during evaluation to yield a model which represents the policy. θ_r and θ_g are optimized by BO. Right: Inferring $\mathbf{a}^\alpha$ given $\mathbf{s}^\alpha$.

We term this approach a *higher-order model* (HOM) which generates (GEN) the model using instructions and parameters into a policy model during evaluation. This HOM is separated into role assignment, and role interaction stages. We visualize an overview of this approach in Fig. 2, left. These parameters are interpreted in context of the current state by the instructions (Alg. 1, Alg. 2) of the HOM to form the policy model which dictates the resultant action.

In our work, each HOM component of role assignment and role interaction is implemented as a neural network.

4.2 Role assignment

Following the success of role based collaboration in multi-agent systems, we assume the interaction and decision making of each agent is governed by its assigned role. Although role based collaboration comes in many forms, we assume[4] that an optimal policy can be decomposed as follows:

$$\pi(\mathbf{a}^1, \ldots, \mathbf{a}^n \mid \mathbf{s}^1, \ldots, \mathbf{s}^n) \triangleq \pi_r(\mathbf{a}^{\alpha(1)}, \ldots, \mathbf{a}^{\alpha(n)} \mid \mathbf{s}^{\alpha(1)}, \ldots \mathbf{s}^{\alpha(n)}) \tag{2}$$

where α is a permutation function dependent on the state, $\mathbf{s}^1, \ldots, \mathbf{s}^n$. The above assumption requires a permutation of agents into roles. For example, in drone delivery, roles could be short-distance deliveries, and long-distance deliveries. In filling these roles, the state of each of the agents are considered. E.g., a drone with low battery may be limited to only performing short-distance deliveries.

To capture this behavior, we define a per role affinity function: $\Lambda^{\theta_{r,i}}(\cdot)$ which is the affinity to take on role i and is parameterized by $\theta_{r,i}$. This function evaluates the affinity of agent ℓ taking on role i using the state of agent: $\mathbf{s}^\ell$. The optimal permutation maximizes the total affinity of an assignment: $\sum_{i=1}^n \Lambda^{\theta_{r,i}}(\mathbf{s}^{\alpha(i)})$ where α represents a permutation. This problem can be efficiently solved using the Hungarian algorithm. We integrate the Hungarian algorithm in our HOM approach during the GEN process. We formalize this in Algorithm 1 which forms the instructions in the role assignment HOM.

Given Algorithm 1, during GEN process, the agents' state, $\mathbf{s}^1, \ldots, \mathbf{s}^n$ is contextually interpreted to yield a permutation model: α. Going forward, we consider the problem of determining the joint policy $\pi_r(\mathbf{a}^{\alpha(1)}, \ldots, \mathbf{a}^{\alpha(n)} \mid \mathbf{s}^{\alpha(1)}, \ldots \mathbf{s}^{\alpha(n)})$ which enables collaborative interactions.

[3]This approach to efficiency is similar in spirit to the work of Lee et al. (1986).
[4]This is a common assumption in multi-agent systems, see, e.g., Le et al. (2017b).

Algorithm 1 *RoleAssignment*

Require: $\mathbf{s}^1, \ldots, \mathbf{s}^n$

return $\arg\max_\alpha \sum_{i=1}^{n} \Lambda^{\theta_{r,i}}(\mathbf{s}^{\alpha(i)})$

Algorithm 4 DSS-GP-UCB

Require: v, H, k

for $t \leftarrow 1, \ldots, T_0$ **do**
- $\theta_{t,h} \sim \mathcal{U}(\Theta)$
- **for** $\ell \leftarrow 1, \ldots, C_1$ **do** $h_{t,\ell} \leftarrow H(\theta_{t,h})$

$\widetilde{E}_d \leftarrow \left|\sum h\right| > c_h; \widetilde{\mathcal{G}}_d \leftarrow (\{\Theta^1, \ldots, \Theta^D\}, \widetilde{E}_d)$

$[\Theta^{(i)}]_{i=1,\ldots,M} \leftarrow$ Max-Cliques$(\widetilde{\mathcal{G}}_d)$; $k \leftarrow \sum_{i=1}^{M} k^{\Theta^{(i)}}$

for $t \leftarrow T_0, \ldots, T$ **do**
- $\theta_t \leftarrow \arg\max_\theta \mu_{t-1}^k(\theta) + \sqrt{\beta_t}\sigma_{t-1}^k(\theta)$
- Query θ_t to observe $y_t = v(\theta_t) + \mathcal{N}(0, \epsilon^2)$
- Update posterior, μ, σ, with θ_t, y_t

Algorithm 2 *RoleInteraction*

Require: $\mathbf{s}^{\alpha(1)}, \ldots, \mathbf{s}^{\alpha(n)}$

for $i \leftarrow 1, \ldots, n$ **do**
- **for** $\ell \leftarrow 1, \ldots, n$ **do** ▷ Edge affinities.
  - **if** $\Lambda^{\theta_{g,v}}(\mathbf{s}^{\alpha(i)}, \mathbf{s}^{\alpha(\ell)}) > 0$ **then**
    - $N^{\alpha(i)}.append(\alpha(\ell))$

return $N^{\alpha(1)}, \ldots, N^{\alpha(n)}$

Algorithm 3 GEN-*Policy*

Require: $\mathbf{s}^1, \ldots, \mathbf{s}^n$

$\alpha \leftarrow RoleAssignment(\mathbf{s}^1, \ldots, \mathbf{s}^n)$

$N \leftarrow RoleInteraction(\mathbf{s}^{\alpha(1)}, \ldots, \mathbf{s}^{\alpha(n)})$

$\mathbf{a} \leftarrow \text{MPNN}(\mathbf{s}^\alpha, N)$ ▷ See Eq. 3

return $[a^{\alpha^{-1}(i)}]_{i=1,\ldots,n}$ ▷ Invert permutation.

4.3 Role interaction

Capturing multiple roles working together is an important part of an effective multi-agent policy. For example in drone delivery, drones must both divide the available task among themselves, as well as use collision avoidance while executing deliveries. Modeling role interactions must accomplish two goals. Firstly, agent interactions may change over time. For example collision avoidance strategies involve the closest drones which change as the drone moves within the environment. Secondly, efficient parameterization is needed as the number of interactions scales *quadratically* due to considering interaction between *all pairs of agents*.

To overcome these challenges, we propose a HOM which generates (GEN) a graphical model. The GEN process is conditioned on the agents' state, thus capturing dynamic role interactions; in addition the GEN process allows for a more compact policy space with far fewer parameters. The resultant generated graphical model captures the state-dependent interaction between roles and yields the resultant actions for each role. After GEN, the interaction between roles are captured by the resultant conditional random field. This is presented in Fig. 2, right. The MRF (Markov Random Field) represents arbitrary undirected connectivity between nodes $\mathbf{a}^{\alpha(1)}, \ldots, \mathbf{a}^{\alpha(n)}$, which is denoted by $\mathcal{G}$. This connectivity allows different roles to collaborate together to determine the joint action.[5]

We perform inference over the graphical model presented in Fig. 2 using Message Passing Neural Networks (Gilmer et al., 2017) (MPNN). We present iterative message passing rules to map from $\mathbf{s}^\alpha$ to a^α:

$$m_{t+1}^{\alpha(i)} \triangleq \sum_{\alpha(\ell) \in N^{\alpha(i)}} M^{\theta_{g,\eta}}\left(h_t^{\alpha(i)}, h_t^{\alpha(\ell)}, i, \ell\right); \quad h_{t+1}^{\alpha(i)} \triangleq U^{\theta_{g,e}}\left(\mathbf{s}^{\alpha(\mathbf{i})}, h_t^{\alpha(i)}, m_{t+1}^{\alpha(i)}\right); \quad \mathbf{a}^\alpha \triangleq \left[h_\tau^{\alpha(i)}\right]_{i=1,\ldots,n} \tag{3}$$

where M is the message function parameterized by $\theta_{g,\eta}$, U is the action update function parameterized by $\theta_{g,e}$, $N^{\alpha(i)}$ denotes the neighbors of $\alpha(i)$. This procedure concludes after τ iterations of message passing with the policy actions indicated by the hidden states, $\left[h_\tau^{\alpha(i)}\right]_{i=1,\ldots,n}$.

To generate graphical models of the above form, our HOM uses edge affinity functions. This approach overcomes the quadratic scaling in modeling all pairs of interaction. Edge affinity functions $\Lambda^{\theta_{g,v}}(\cdot)$ determine whether an edge exists between node $\mathbf{a}^{\alpha(i)}$, and $\mathbf{a}^{\alpha(\ell)}$. The graphical model GEN process is presented in Algorithm 2. Finally, Algorithm 3 drives the GEN process.

4.4 Additive decomposition

Although our HOM policy representation is compact, it is still of significant dimensionality which makes optimization with BO difficult. HDBO is challenging due to the curse of dimensionality with common kernels

[5]We refer readers to Wang et al. (2013) for additional overview.

such as Matern or RBF.[6] A common technique to overcome this is through assuming additive structural decomposition on v: $v(\theta) \triangleq \sum_{i=1}^{M} v^{(i)}(\theta^{(i)})$ where $v^{(i)}$ are independent functions, and $\theta^{(i)} \in \Theta^{(i)}$ (Duvenaud et al., 2011). Specifically $\Theta \triangleq \Theta^1 \times \ldots \times \Theta^D$ for some dimensionality D, and $\Theta^{(i)} \subseteq \{\Theta^1, \ldots, \Theta^D\}$ and is of low dimensionality. This structural assumption is combined with the assumption that each $v^{(i)}$ is sampled from a GP. If $v^{(i)} \sim \mathrm{GP}\big(0, k^{\Theta^{(i)}}(\theta^{(i)}, \theta^{(i)'})\big)$ then $v \sim \mathrm{GP}\big(0, \sum_i k^{\Theta^{(i)}}(\theta^{(i)}, \theta^{(i)'})\big)$ (Rasmussen & Williams, 2006). This assumption decomposes a high dimensional GP surrogate model of v into a set of many low dimensional GPs, which is easier to jointly learn and optimize.

An additive decomposition can be represented by a dependency graph between the dimensions: $\mathcal{G}_d \triangleq (V_d, E_d)$ where $V_d \triangleq \{\Theta^1, \ldots, \Theta^D\}$ and $E_d \triangleq \{(\Theta^a, \Theta^b) \mid a, b \in \Theta^{(i)} \text{ for some } i\}$. *We **highlight** that this graph is between the dimensions of the policy parameters, Θ, and is unrelated to the graphical model of role interactions presented in earlier sections.* It is possible to accurately model v by a kernel $k \triangleq \sum_i k^{\Theta^{(i)}}$ where each $\Theta^{(i)}$ corresponds to a *maximal clique* of the dependency graph (Rolland et al., 2018). Knowing the dependency graph greatly simplifies the complexity of optimizing v.

However, learning the dependency graph in additive decomposition remains challenging as there are $O(D^2)$ possible edges each of which may be present or absent yielding $2^{O(D^2)}$ possible dependency structures. This difficult problem is often approached using inefficient stochastic sampling methods such as Gibbs sampling.

4.5 Dependency Structure Search Bayesian Optimization

We propose learning the dependency structure during the GEN process. Our approach is based on the following observation which is illustrated in Fig. 1:

Proposition 1. *Let $\mathcal{G}_d = (V_d, E_d)$ represent an additive dependency structure with respect to $v(\theta)$, then the following holds true: $\forall a, b \ \frac{\partial^2 v}{\partial \theta^a \partial \theta^b} \neq 0 \implies (\Theta^a, \Theta^b) \in E_d$ which is a consequence of v formed through addition of independent sub-functions $v^{(i)}$, at least one of which must contain θ^a, θ^b as parameters for $\frac{\partial^2 v}{\partial \theta^a \partial \theta^b} \neq 0$ which implies their connectivity within E_d.*

Following this, we consider algorithms with noisy query access to the Hessian, $\mathbf{H}_v$.

Assumption 1. *Let $\mathcal{G}_d = (V_d, E_d)$ be sampled from an Erdős-Rényi model with probability $p_g < 1$: $\mathcal{G}_d \sim G(D, p_g)$. That is, each edge (Θ^a, Θ^b) is i.i.d. sampled from a binomial distribution with probability, p_g. With $[\Theta^{(i)}]_{i=1,\ldots,M}$ representing the maximal cliques of $\mathcal{G}_d$, we assume that $v \sim GP\left(0, \sum_i k^{\Theta^{(i)}}(\theta^{(i)}, \theta^{(i)'})\right)$ for some kernel k taking an arbitrary number of arguments (e.g., RBF). Noisy queries can be made to the Hessian of v, $\mathbf{H}_v$. We define $H(\theta) \triangleq [\frac{\partial^2 v}{\partial \theta^a \partial \theta^b} + \epsilon_h^{(a,b)}]_{a,b=1,\ldots,D}$ where $\epsilon_h^{(a,b)} \sim \mathcal{N}(0, \sigma_n^2)$ i.i.d. Each query to H has corresponding regret of $r(\theta)$.*

Under this set of assumptions, we present DSS-GP-UCB in Algorithm 4. DSS-GP-UCB follows the overall structure of GP-UCB with two additions. We perform C_1 queries to the Hessian if $t \leq T_0$. These Hessian queries are then averaged and compared to a cutoff constant c_h to determine the dependency structure $\widetilde{E}_d$. After extraction of maximal cliques depending on $\widetilde{E}_d$ we construct $k = \sum_i k^{\Theta^{(i)}}$, the sum of the aforementioned kernels and inference and acquisition proceeds same as GP-UCB.

To bound the cumulative regret, $R_t \triangleq \sum_{t=1}^{T_0} C_1 r(\theta_{t,h}) + \sum_{t=T_0}^{T} r(\theta_t)$, we show that after $C_1 T_0$ queries to the Hessian, with high probability we have $\widetilde{E}_d = E_d$, where E_d is the unknown ground truth dependency structure for v.

Theorem 1. *Suppose[7] there exists σ_h^2, p_h s.t. $\forall i, j \ \mathbb{P}_{\theta \sim \mathcal{U}(\Theta)}\left[k^{\partial i \partial j}(\theta, \theta) \geq \sigma_h^2\right] \geq p_h$ and $\forall i, j, \theta, \theta'$ $k^{\partial i \partial j}(\theta, \theta') \geq 0$. Then for any $\delta_1, \delta_2 \in (0, 1)$ after $t \geq T_0$ steps of DSS-GP-UCB we have: $\bigcap_{i,j} P(\widetilde{E}_d^{i,j} = E_d^{i,j}) \geq 1 - \delta_1 - \delta_2$ when $T_0 = C_1 > \frac{8D^2}{\delta_1^2} \log \frac{2D^2}{\delta_1} \frac{\sigma_n^2}{\sigma_h^2} + \frac{D^2}{p_h \delta_2}$, $c_h \triangleq T_0 \sigma_n \sqrt{2 \log \frac{2D^2}{\delta_1}}$.*

[6] A parallel area in HDBO is of computational efficiency of acquisition which is outside the scope of this work. We refer readers to the works of Mutny & Krause (2018), Wilson et al. (2020), and Ament & Gomes (2022).

[7] RBF kernel satisfies these assumptions when $\Theta = [0, 1]^D$.

Our Theorem 1 relies on repeatedly sampling the Hessian to determine whether an edge exists between Θ^a, and Θ^b in the sampled additive decomposition. The key challenge is determining this connectivity under a very noisy setting, and for extremely low values of $\sigma_h^2 \ll \sigma_n^2$ where the Hessian is zero with high probability. We are able to overcome this challenge using a Bienaymé's identity, a key tool in our analysis. We defer all proofs to the Appendix.

Utilizing the above theorem we are able to provide a regret bound for DSS-GP-UCB. Providing this regret bound requires several key tools. First, we are able to bound the number and size of cliques of graphs sampled from the Erdős-Rényi model with high probability. Secondly, we are able to bound the *mutual information* of an additive decomposition given the mutual information of its constituent kernels using Weyl's inequality. Lastly, we use similar analysis as Srinivas et al. (2010) to complete the regret bound.

Theorem 2. *Let k be the kernel as in Assumption 1, and Theorem 1. Let $\gamma_T^k(d) : \mathbb{N} \to \mathbb{R}$ be a monotonically increasing upper bound function on the* mutual information *of kernel k taking d arguments. The cumulative regret of DSS-GP-UCB is bounded with high probability as follows:*

$$R_T = \widetilde{\mathcal{O}}\Big(\sqrt{T\beta_T D^{\mathcal{O}(\log D)}\gamma_T^k(\mathcal{O}(\log D))}\Big). \tag{4}$$

Whereas for typical kernels such as Matern and RBF, cumulative regret of GP-UCB scales exponentially with D, our regret bounds scale with exponent $\mathcal{O}(\log D)$. This improved regret bound shows our approach is a theoretically grounded approach to HDBO.

In practice, observing the hessian $\mathbf{H}_v$ is not possible due to v being a black box function. However, during the GEN process we can observe a surrogate Hessian, $\mathbf{H}_\pi$. This surrogate Hessian is closely related to the $\mathbf{H}_v$ as $v(\theta)$ is determined through interaction of the policy with an unknown environment. Because the *value* of a policy is a function of the policy; it follows by the chain rule[8] $\mathbf{H}_\pi$ is an important sub-component of $\mathbf{H}_v$. We utilize the surrogate Hessian in our work and demonstrate its strong empirical performance in validation.

5 Validation

We compare our work against recent algorithms in MARL on several multi-agent coordination tasks and RL algorithms for policy search in novel settings. We also perform ablation and investigation of our proposed HOM at learning roles and multi-agent interactions. We defer experimental details to Appendix A.

All presented figures are average of 5 *runs with shading representing* $\pm$ *Standard Error, the y-axis represents cumulative reward, the x-axis displayed above represents interactions with the environment in* l*, x-axis displayed below represents iterations of* BO*. Commensurate with our focus on memory-constrained devices, all policy models consist of* < 500 *parameters.*

5.1 Ablation

We investigate the impact of Role Assignment (RA) and Role Interaction (RI) as well as model capacity on training progress. We conduct ablation experiments on Multiagent Ant with 6 agents, PredPrey with 3 agents, and Heterogenous PredPrey with 3 agents. Multiagent Ant is a MuJoCo locomotion task where each agent controls an individual appendage. PredPrey is a task where predators must work together to catch faster, more agile prey. Het. PredPrey is similar, except the predators have different capabilities of speed and acceleration. In ablation experiments, our default configuration is *Med - RA - RI* which employs components of RA and RI parameterized by neural networks with three layers and four neurons on each layer. We present our ablation in Fig. 3.

For a simpler coordination task such as Multiagent Ant, we observe limited improvement through RA or RI. In contrast, RI shows strong improvement in PredPrey and Het. PredPrey. It is because, in PredPrey, predators must work together to catch the faster prey. Since the agents in PredPrey are *homogeneous*, *ablating RA* makes the optimization simpler and more compact without losing expressiveness. Thus, ablating RA leads to a performance increase. In Het. PredPrey, the predator agents have heterogeneous capabilities

[8]We revisit this argument in Appendix G.

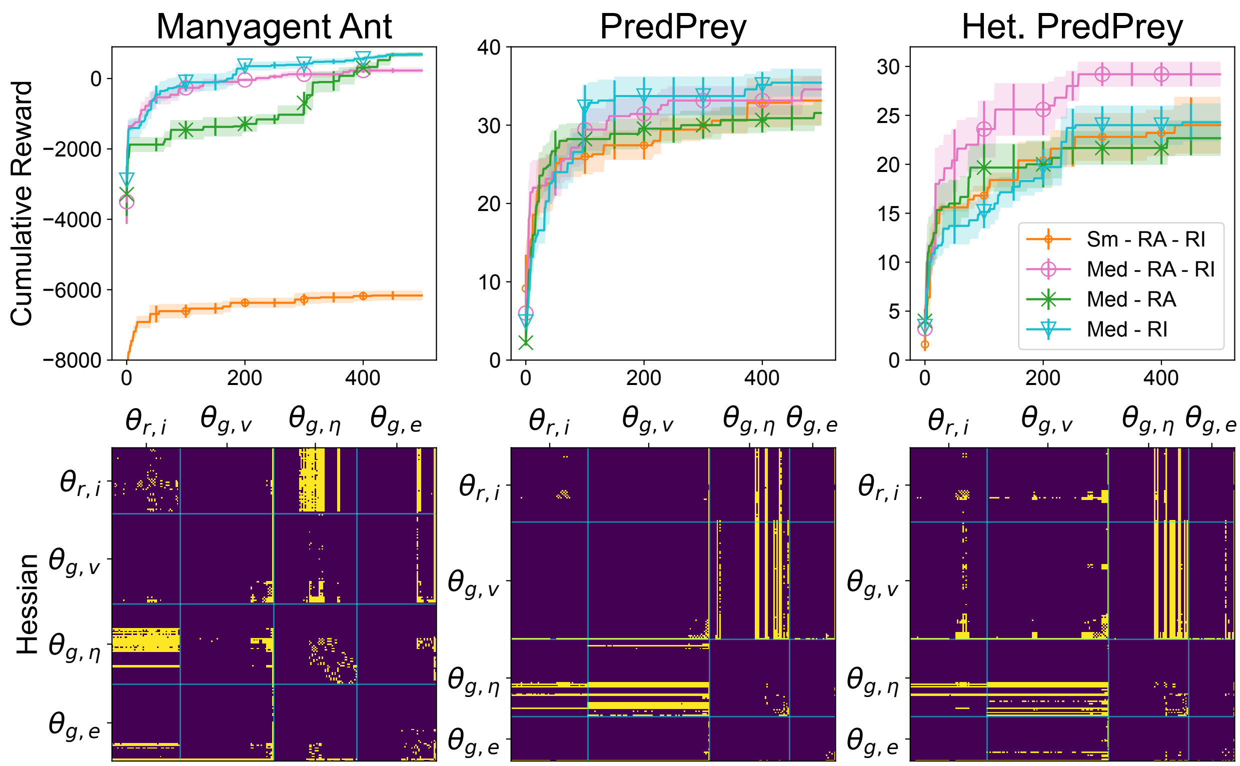

Figure 3: Ablation study. Training curves of our HOM and its ablated variants on different multi-agent environments.

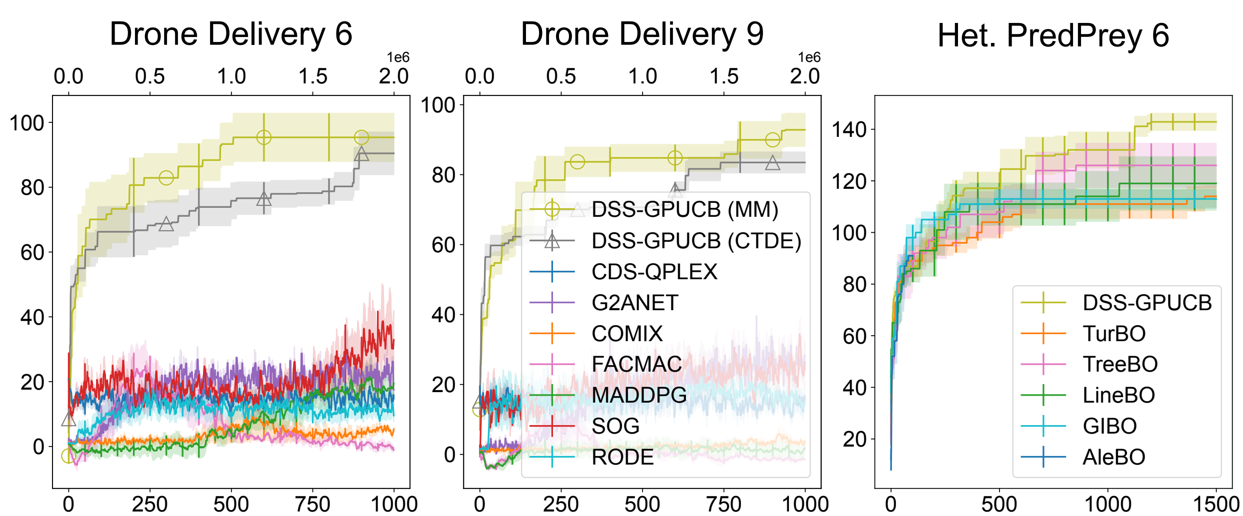

Figure 4: Left: Sparse reward drone delivery task. Right: Comparison with HDBO approaches.

in speed and acceleration. Thus, RA plays a critical role in delivering strong performance. We also show that overly shrinking the model size (*Sm - RA - RI*) can hurt performance as the policy model is no longer sufficiently expressive. This is evidenced in the Multiagent Ant task. We observed that using neural networks of three layers with four neurons each to be sufficiently balanced across a wide variety of tasks.

In Fig. 3, we present the detected Hessian structure by DSS-GP-UCB in the respective tasks. The detected Hessian structures generally show strong block-diagonal associativity in the HOM parameters, i.e.,

Table 2: DSS-GP-UCB typically outperforms RL with higher sparsity (e.g., Sparse-100, or Sparse-200).

	Ant-v3					Hopper-v3					Swimmer-v3					Walker2d-v3				
	DDPG	PPO	SAC	TD3	Intrinsic	DDPG	PPO	SAC	TD3	Intrinsic	DDPG	PPO	SAC	TD3	Intrinsic	DDPG	PPO	SAC	TD3	Intrinsic
Baseline	−90.77	1105.69	2045.24	2606.17	2144.00	604.20	1760.65	2775.66	1895.76	1734.00	44.45	121.38	58.73	48.78	1950.00	2203.80	892.81	4297.03	1664.46	2210.00
Sparse 2	−32.88	1007.80	2563.97	1407.40	1964.00	877.93	1567.14	3380.60	1570.84	2074.00	35.59	99.50	46.75	47.23	1758.80	1470.62	1471.33	1673.46	2297.43	1952.00
Sparse 5	−2687.97	961.31	711.56	762.61	1916.00	814.59	1616.79	3239.20	2290.67	1972.00	26.66	68.69	43.84	40.12	1856.00	961.30	697.93	1697.25	2932.27	1924.00
Sparse 20	−2809.89	624.07	694.30	379.12	1838.00	783.95	1629.28	2535.17	1436.33	1537.20	19.12	54.63	37.78	37.03	2108.00	663.04	365.39	1010.63	276.56	1810.00
Sparse 50	−3067.37	−67.43	663.28	253.66	1091.20	816.25	1010.73	1238.03	551.43	642.00	23.73	51.52	38.78	30.01	812.00	572.12	428.29	349.47	298.28	834.75
Sparse 100	−3323.43	−4021.56	679.30	−115.43	450.40	988.36	324.51	260.52	342.48	406.80	9.64	21.09	27.98	30.10	376.60	523.89	205.93	200.16	147.22	480.60
Sparse 200	−3098.37	−8167.98	−107.14	−147.86	258.60	765.05	222.76	300.36	281.68	350.80	−9.97	21.69	33.35	30.48	342.80	182.84	193.43	187.16	148.06	353.20
DSS-GP-UCB			1147.21					1009.3					175.73					1008.90		

$[\theta_{r,i}, \theta_{g,v}, \theta_{g,\eta}, \theta_{g,e}]$. This shows that our approach can detect the interdependence *within* the sub-parameters, but relative independence between the sub-parameters. We observe more off-diagonal connectivity in the complex coordination tasks of PredPrey and Het. PredPrey. The visualization of Hessian structure on PredPrey shows that our approach can detect the importance of *jointly optimizing* role assignment and interaction to deliver a strong policy in this complex coordination task. We investigate the learning behavior of the HOM further in Appendix B.

5.2 Comparison with MARL

We compare our method with competing MARL algorithms on several multi-agent tasks where the number of agents is increased. We validate both the HOM with DSS-GP-UCB (DSS-GP-UCB (MM)) and neural network policies trained in the CTDE paradigm (DSS-GP-UCB (CTDE)). We observe that on complex coordination tasks such as PredPrey and Het. PredPrey our approach delivers more performant policies when coordination is required between *a large number of agents*. This is presented[9] in Fig. 5. Although SOG (Shao et al., 2022), a Comm-MARL approach shows compelling performance with a small number of agents, with 15 agents, both DSS-GP-UCB (CTDE) and DSS-GP-UCB (MM) outperform this strategy. We highlight that DSS-GP-UCB (CTDE) outperforms Comm-MARL approaches without communication during execution. We also note that DSS-GP-UCB (MM) outperforms DSS-GP-UCB (CTDE) showing the value of our HOM approach in complex coordination tasks. We defer further experimental results in this setting to Appendix B.

5.3 Policy optimization under malformed reward

We compare against several competing RL and MARL algorithms under malformed reward scenarios. We train neural network policies with DSS-GP-UCB and competing algorithms. We consider a sparse reward scenario where reward feedback is given every S environment interactions for varying S. Table 2 shows that the performance of competing algorithms is severely degraded with sparse reward and DSS-GP-UCB outperforms competing approaches on most tasks with moderate or higher sparsity. Although intrinsic motivation (Singh et al., 2004; Zheng et al., 2018) has shown evidence in overcoming this limitation, we find that our approach outperforms competing approaches supported by intrinsic motivations at higher sparsity. This improvement is important as sparse and malformed reward structure scenarios can occur in real-world tasks (Aubret et al., 2019). We repeat this validation in Appendix B with MARL algorithms in multi-agent settings and consider a delayed feedback setting with similar results.

5.4 Higher-order model Investigation

We examined policy for Multiagent Ant with 6 agents for the role based policy specialization. The policy modulation plots were generated by examining the PredPrey and Het. PredPrey environments respectively.

In Fig. 6 we investigate the learned HOM policies. Our investigation shows that *role* is used to specialize agent policies while maintaining a common theme. *Role interaction* modulates the policy through graphical model inferences. Finally, role interactions are sparse, however noticeably higher for complex coordination tasks such as PredPrey.

[9]We plot with respect to total environment interactions for l, and total policy evaluations for BO. See Appendix I, Appendix J, and Appendix K for alternate presentations of data more favorable to RL and MARL under which our conclusions still hold.

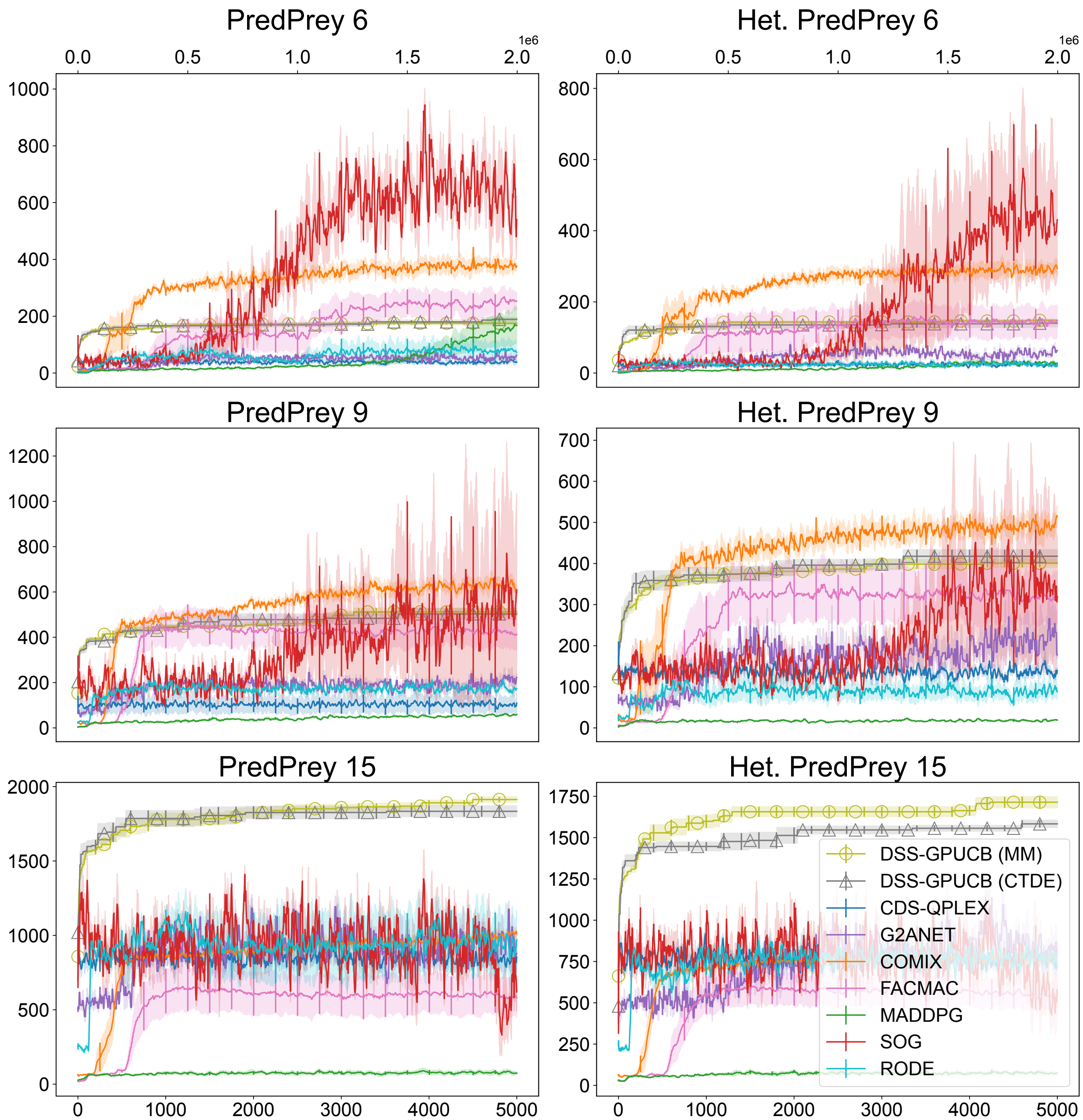

Figure 5: Scaling analysis. Training curves of DSS-GP-UCB and competitors with increasing number of agents. The left column shows PredPrey with 6, 9, and 15 agents. The right column shows Het, PredPrey with 6, 9, and 15 agents.

5.5 Comparison with HDBO algorithms

We compare with several related work in HDBO. This is presented in Fig. 4. We compare against these algorithms at optimizing our HOM policy. For more complex tasks that require role based interaction and coordination, our approach outperforms related work. TreeBO (Han et al., 2021) is also an additive decomposition approach to HDBO, but uses Gibbs sampling to learn the dependency structure. However, our approach of learning the structure through *Hessian-Awareness* outperforms this approach. Additional experimental results are deferred to Appendix B.

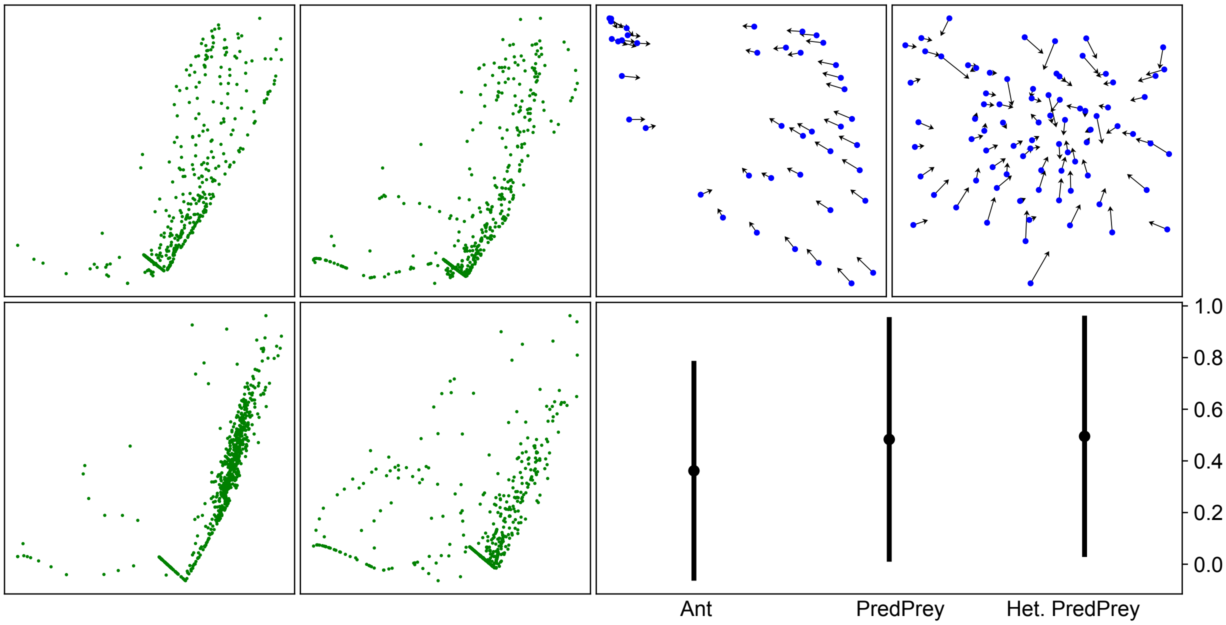

Figure 6: Left: Action distributions of different roles showing diversity in the Multiagent Ant environment with 6 agents. Right above: Policy modulation with role interaction in PredPrey and Het. PredPrey environment with 3 agents. Arrows represent change after message passing. Right below: Mean connectivity and standard deviation in role interaction in Multiagent Ant with 6 agents, PredPrey with 3 agents, and Het. PredPrey with 3 agents.

5.6 Drone delivery task

We design a drone delivery task that is well aligned with our motivation of considering policy search in *memory-constrained devices* on tasks with *unhelpful or noisy gradient information*. In this task, drones must maximize the throughput of deliveries while avoiding collisions and conserving fuel. This task is challenging as a positive reward through completing deliveries is rarely encountered (i.e., sparse rewards). However, agents often receive negative rewards due to collisions or running out of fuel. Thus, gradient-based approaches can easily fall into local minima and fail to find policies that complete deliveries.[10] We compare DSS-GP-UCB against competing approaches in Fig. 4. We observe that MARL based approaches fail to find a meaningfully rewarding policy in this setting, whereas our approach shows strong and compelling performance. Furthermore, DSS-GP-UCB (MM) outperforms DSS-GP-UCB (CTDE) through leveraging roles and role interactions.

6 Conclusion

We have proposed a HOM policy along with an effective optimization algorithm, DSS-GP-UCB. Our HOM and DSS-GP-UCB are designed to offer strong performance in high coordination multi-agent tasks under sparse or malformed reward on memory-constrained devices. DSS-GP-UCB is a theoretically grounded approach to BO offering good regret bounds under reasonable assumptions. Our validation shows DSS-GP-UCB outperforms RL and MARL at optimizing neural network policies in malformed reward scenarios. Our HOM optimized with DSS-GP-UCB outperforms MARL approaches in high coordination multi-agent scenarios by leveraging the concepts of *role* and *role interaction*. Furthermore, we show through our drone delivery task, our approach outperforms MARL approaches in multi-agent coordination tasks with sparse reward. We make significant progress on high coordination multi-agent policy search by overcoming challenges posed by malformed reward and memory-constrained settings.

[10] Further details on this task can be found in Appendix H.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Riad Akrour, Dmitry Sorokin, Jan Peters, and Gerhard Neumann. Local bayesian optimization of motor skills. In *Proc. ICML*, 2017.

Sebastian E. Ament and Carla P. Gomes. Scalable first-order bayesian optimization via structured automatic differentiation. In *Proc. ICML*, pp. 500–516, 2022.

Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.*, 34(6):26–38, 2017.

Arthur Aubret, Laëtitia Matignon, and Salima Hassas. A survey on intrinsic motivation in reinforcement learning. *CoRR*, abs/1908.06976, 2019.

Andrei-Cristian Barbos, Francois Caron, Jean-François Giovannelli, and Arnaud Doucet. Clone MCMC: parallel high-dimensional gaussian gibbs sampling. In *Proc. NeurIPS*, 2017.

Joel Berkeley, Henry B. Moss, Artem Artemev, Sergio Pascual-Diaz, Uri Granta, Hrvoje Stojic, Ivo Couckuyt, Jixiang Qing, Nasrulloh Loka, Andrei Paleyes, Sebastian W. Ober, and Victor Picheny. Trieste, 7 2022. URL `https://github.com/secondmind-labs/trieste`.

Béla Bollobás and Paul Erdös. Cliques in random graphs. In *Proc. Cambridge Philosophical Society*, 1976.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Shuyu Cheng, Guoqiang Wu, and Jun Zhu. On the convergence of prior-guided zeroth-order optimization algorithms. In *Proc. NeurIPS*, pp. 14620–14631, 2021.

John T Chu. On bounds for the normal integral. *Biometrika*, 42:263–265, 1955.

Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. In *Proc. ICML*, pp. 1538–1546, 2019.

Christian Schroeder de Witt, Bei Peng, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. Deep multi-agent reinforcement learning for decentralized continuous cooperative control. *arXiv preprint arXiv:2003.06709*, 2020.

Carlo D'Eramo, Davide Tateo, Andrea Bonarini, Marcello Restelli, and Jan Peters. Mushroomrl: Simplifying reinforcement learning research. 2021.

Kevin Dorling, Jordan Heinrichs, Geoffrey G. Messier, and Sebastian Magierowski. Vehicle routing problems for drone delivery. *IEEE Trans. Syst. Man Cybern. Syst.*, 47(1):70–85, 2017.

David Duvenaud, Hannes Nickisch, and Carl Edward Rasmussen. Additive gaussian processes. In *Proc. NeurIPS*, pp. 226–234, 2011.

David Eriksson, Michael Pearce, Jacob R. Gardner, Ryan Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. In *Proc. NeurIPS*, 2019a.

David Eriksson, Michael Pearce, Jacob R. Gardner, Ryan Turner, and Matthias Poloczek. Scalable global optimization via local Bayesian optimization. In *Proc. NeurIPS*, pp. 5497–5508, 2019b.

Natalia Y. Ermolova and Sven-Gustav Häggman. Simplified bounds for the complementary error function. In *Proc. Eurasip*, pp. 1087–1090, 2004.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Lukas P. Fröhlich, Melanie N. Zeilinger, and Edgar D. Klenske. Cautious bayesian optimization for efficient and scalable policy search. In *Proc. L4DC*, 2021.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proc. ICML*, pp. 1263–1272, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Eric Han, Ishank Arora, and Jonathan Scarlett. High-dimensional bayesian optimization via tree-structured additive models. *arXiv preprint arXiv:2012.13088*, 2020.

Eric Han, Ishank Arora, and Jonathan Scarlett. High-dimensional Bayesian optimization via tree-structured additive models. In *Proc. AAAI*, pp. 7630–7638, 2021.

Verena Heidrich-Meisner and Christian Igel. Evolution strategies for direct policy search. In *Proc. PPSN*, pp. 428–437, 2008.

Matthew J. Johnson, James Saunderson, and Alan S. Willsky. Analyzing hogwild parallel gaussian gibbs sampling. In *Proc. NeurIPS*, 2013.

Kirthevasan Kandasamy, Jeff G. Schneider, and Barnabás Póczos. High dimensional bayesian optimisation and bandits via additive models. In *Proc. ICML*, pp. 295–304, 2015.

Johannes Kirschner, Mojmir Mutny, Nicole Hiller, Rasmus Ischebeck, and Andreas Krause. Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces. In *Proc. ICML*, pp. 3429–3438, 2019.

Hoang M Le, Yisong Yue, Peter Carr, and Patrick Lucey. Coordinated multi-agent imitation learning. In *International Conference on Machine Learning*, pp. 1995–2003. PMLR, 2017a.

Hoang Minh Le, Yisong Yue, Peter Carr, and Patrick Lucey. Coordinated multi-agent imitation learning. In *Proc. ICML*, pp. 1995–2003, 2017b.

YC Lee, Gary Doolen, HH Chen, GZ Sun, Tom Maxwell, and HY Lee. Machine learning using a higher order correlation network. Technical report, Los Alamos National Lab (LANL), Los Alamos, NM (United States); Univ. of Maryland, College Park, MD (United States), 1986.

Benjamin Letham, Roberto Calandra, Akshara Rai, and Eytan Bakshy. Re-examining linear embeddings for high-dimensional Bayesian optimization. In *Proc. NeurIPS*, 2020.

Kemas M Lhaksmana, Yohei Murakami, and Toru Ishida. Role-based modeling for designing agent behavior in self-organizing multi-agent systems. *International Journal of Software Engineering and Knowledge Engineering*, 28(01):79–96, 2018.

Chenghao Li, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. Celebrating diversity in shared multi-agent reinforcement learning. In *Proc. NeurIPS*, pp. 3991–4002, 2021.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Yong Liu, Weixun Wang, Yujing Hu, Jianye Hao, Xingguo Chen, and Yang Gao. Multi-agent game abstraction via graph attention neural network. In *Proc. AAAI*, pp. 7211–7218, 2020.

Daniel J. Lizotte, Tao Wang, Michael H. Bowling, and Dale Schuurmans. Automatic gait optimization with gaussian process regression. In *Proc. IJCAI*, 2007.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Eduardo Magalhães. On the properties of the hessian tensor for vector functions. *viXra preprint viXra:2005.0044*, 2020.

Alonso Marco, Philipp Hennig, Jeannette Bohg, Stefan Schaal, and Sebastian Trimpe. Automatic LQR tuning based on gaussian process global optimization. In *Proc. ICRA*, 2016.

Ruben Martinez-Cantin. Bayesian optimization with adaptive kernels for robot control. In *Proc. ICRA*, 2017.

Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, apr 2017. URL `http://jmlr.org/papers/v18/16-537.html`.

David W Matula. *The largest clique size in a random graph.* Department of Computer Science, Southern Methodist University Dallas, Texas, 1976.

Mark McLeod, Stephen J. Roberts, and Michael A. Osborne. Optimization, fast and slow: optimally switching between local and bayesian optimization. In *Proc. ICML*, 2018.

Massimo Merenda, Carlo Porcaro, and Demetrio Iero. Edge machine learning for AI-Enabled IoT devices: A review. *Sensors*, 20(9):2533, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Sarah Müller, Alexander von Rohr, and Sebastian Trimpe. Local policy search with Bayesian optimization. In *Proc. NeurIPS*, pp. 20708–20720, 2021.

Mojmir Mutny and Andreas Krause. Efficient high dimensional bayesian optimization with additivity and quadrature fourier features. In *Proc. NeurIPS*, pp. 9019–9030, 2018.

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.

Evgenia Papavasileiou, Jan Cornelis, and Bart Jansen. A systematic literature review of the successors of "neuroevolution of augmenting topologies". *Evolutionary Computation*, 29(1):1–73, 2021.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. Facmac: Factored multi-agent centralised policy gradients. *Advances in Neural Information Processing Systems*, 34:12208–12221, 2021.

Peng Peng, Ying Wen, Yaodong Yang, Quan Yuan, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017.

Victor Picheny, Henry B. Moss, Léeonard Torossian, and Nicolas Durrande. Bayesian quantile and expectile optimisation. In *Proc. UAI*, pp. 1623–1633, 2022.

Hong Qian and Yang Yu. Derivative-free reinforcement learning: a review. *Frontiers of Computer Science*, 15 (6):1–19, 2021.

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.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning.* MIT Press, 2006.

Yara Rizk, Mariette Awad, and Edward W Tunstel. Decision making in multiagent systems: A survey. *IEEE Transactions on Cognitive and Developmental Systems*, 10(3):514–529, 2018.

Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

Paul Rolland, Jonathan Scarlett, Ilija Bogunovic, and Volkan Cevher. High-dimensional bayesian optimization via additive models with overlapping groups. In *Proc. AISTATS*, pp. 298–307, 2018.

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

Guy Shani, Joelle Pineau, and Robert Kaplow. A survey of point-based POMDP solvers. *Autonomous Agents and Multi-Agent Systems*, 27:1–51, 2013.

Jianzhun Shao, Zhiqiang Lou, Hongchang Zhang, Yuhang Jiang, Shuncheng He, and Xiangyang Ji. Self-organized group for cooperative multi-agent reinforcement learning. *Proc. NeurIPS*, pp. 5711–5723, 2022.

Shubhanshu Shekhar and Tara Javidi. Significance of gradient information in bayesian optimization. In *Proc. AISTATS*, 2021.

Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. In *Proc. ICLR*, 2019.

Satinder Singh, Andrew G. Barto, and Nuttapong Chentanez. Intrinsically motivated reinforcement learning. In *Proc. NeurIPS*, pp. 1281–1288, 2004.

Maciej Skorski. Chain rules for hessian and higher derivatives made easy by tensor calculus. *arXiv preprint arXiv:1911.13292*, 2019.

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.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. ICML*, 2010.

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. *arXiv preprint arXiv:1706.05296*, 2017.

Alexander von Rohr, Sebastian Trimpe, Alonso Marco, Peer Fischer, and Stefano Palagi. Gait learning for soft microrobots controlled by light fields. In *Proc. IROS*, 2018.

Chaohui Wang, Nikos Komodakis, and Nikos Paragios. Markov random field modeling, inference & learning in computer vision & image understanding: A survey. *Comput. Vis. Image Underst.*, 117(11):1610–1627, 2013.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. QPLEX: Duplex dueling multi-agent Q-Learning. In *Proc. ICLR*, 2021a.

Linnan Wang, Rodrigo Fonseca, and Yuandong Tian. Learning search space partition for black-box optimization using monte carlo tree search. In *Proc. NeurIPS*, 2020a.

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. In *Proc. ICLR*, 2021b.

Daan Wierstra, Tom Schaul, Jan Peters, and Jürgen Schmidhuber. Fitness expectation maximization. In *Proc. PPSN*, pp. 337–346, 2008.

Aaron Wilson, Alan Fern, and Prasad Tadepalli. Using trajectory data to improve bayesian optimization for reinforcement learning. *JMLR*, 15(1), 2014.

James T. Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Efficiently sampling functions from Gaussian process posteriors. In *Proc. ICML*, pp. 10292–10302, 2020.

Zeyu Zheng, Junhyuk Oh, and Satinder Singh. On learning intrinsic rewards for policy gradient methods. In *Proc. NeurIPS*, pp. 4649–4659, 2018.

Changxi Zhu, Mehdi Dastani, and Shihan Wang. A survey of multi-agent reinforcement learning with communication. *arXiv preprint arXiv:2203.08975*, 2022.